UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UNA RED SOCIAL DE EVENTOS PARA ANDROID

Autor: Eduardo de Quinto Barbado
Tutor: Alejandro Sierra Urrecho

mayo 2019

Eduardo de Quinto Barbado
DISEÑO E IMPLEMENTACIÓN DE UNA RED SOCIAL DE EVENTOS PARA ANDROID

Eduardo de Quinto Barbado

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mi familia y mi pareja por el apoyo en esta nueva etapa de mi vida.
En especial a mi hermana, sin la cual no existiría este proyecto.

Un poco más de persistencia, un poco más de esfuerzo,
y lo que parecía irremediablemente un fracaso
puede convertirse en un éxito glorioso.

Elbert Hubbard

Resumen

En la actualidad uno de los productos software más ampliamente utilizado por los usuarios de Internet son las Redes Sociales. Podemos encontrar muchos tipos de redes sociales enfocadas hacia diferentes aspectos de la vida cotidiana de las personas: vida personal, trabajo, relaciones sentimentales, etc.

En este trabajo se propone una Red Social nativa de Android centrada en los eventos personales. El objetivo de la aplicación es permitir a los usuarios crear, organizar, gestionar y compartir información y archivos multimedia entre los asistentes de los eventos que creen en la aplicación.

A lo largo de la memoria, se ahondará en el proceso de diseño, desarrollo, implementación y pruebas de la aplicación, proporcionando al lector un conocimiento exhaustivo sobre todas las fases del proyecto.

Más concretamente, en esta memoria se pretende que el lector pueda observar toda la funcionalidad que ofrece la aplicación y el proceso de desarrollo de la misma. Así mismo, se expondrán otros aspectos relevantes no funcionales de la aplicación, como por ejemplo, eficiencia y rendimiento. Para dar una visión lo más realista posible de la viabilidad de la aplicación como producto software en el mercado, se expondrán los resultados obtenidos en una prueba de campo con usuarios reales y las experiencias de usuario recogidas.

En líneas generales, se ha conseguido crear un producto software competitivo en el mercado actual de las Redes Sociales, con potencial para convertirse en una herramienta útil en la organización de eventos personales.

Palabras clave

Redes Sociales, Eventos, Android, Aplicación, Diseño, Implementación

Abstract

Currently, one of the software products most widely used by Internet users are Social Networks. We can find many types of social networks focused on different aspects of people's daily life: such as personal life, work or relationships.

In this work, I propose a native Android Social Network focused on personal events. The purpose of this application is to allow users to create, organize, manage and share information and multimedia files among the participants of the events they create in the application.

Throughout the memory, we will delve into the process of designing, developing, implementing and testing of the application, providing the reader with exhaustive knowledge on all the phases of the project.

More specifically, the intention of this report is to perform an analysis of the application's functionality and its developing process. Likewise, other relevant non-functional aspects of the application will be exposed, such as efficiency and performance. To give a vision as realistic as possible of the viability of the application as software product in the market, I will show the results obtained in a field test with real users and the user experiences collected.

In general terms, it has been possible to create a competitive software product in the current market of Social Networks, with the potential to become a useful tool in the organization of personal events.

Keywords

Social Network's, Events, Android, App, Design, Implementation

Índice

1 Introducción **1**

1.1 Motivación 1

1.2 Objetivos 3

1.3 Organización de la memoria 3

2 Estado del arte **5**

3 Diseño **9**

3.1 Funcionalidad 9

3.1.1 Definición de Sistemas 9

3.1.2 Requisitos Funcionales 11

3.1.3 Requisitos No Funcionales 15

3.2 Modelo 16

3.2.1 Diseño de clases 16

3.3 Diseño Visual y Conceptual 17

4 Desarrollo **19**

4.1 Planificación 19

4.2 Aspectos Generales 21

4.2.1 Base de datos 23

4.3 Desarrollo de la Beta 24

4.3.1 Desarrollo del Subsistema de Creación y Gestión de Eventos 25

4.3.2 Desarrollo de los Subsistemas de Mensajería y Fotografías 26

4.3.3 Desarrollo del Subsistema de Localización 28

4.4 Desarrollo del Sistema Final 30

4.4.1 Desarrollo de los Subsistemas de Registro, Inicio de Sesión y Gestión de Perfiles . 30

4.4.2 Desarrollo del Sistema de Gestión de Participantes 33

5 Pruebas y resultados **35**

6 Conclusiones y trabajo futuro **39**

6.1 Conclusiones 39

6.2 Trabajo futuro 39

Bibliografía **41**

Definiciones **43**

Acrónimos 45

Apéndices 47

A Cálculo de Puntos de Función 49

B Base de Datos de la Aplicación 55

C Manual de Usuario 59

Listas

Lista de códigos

4.1 Transición entre Fragmentos . . . 25
4.2 Transición entre Fragmentos . . . 25
4.3 Creación de Thumbnails . . . 27
4.4 Compresión de imágenes . . . 28
4.5 Ubicacion GoogleMaps . . . 29
4.6 Registro de un usuario en Firebase Authentication . . . 31
4.7 Autenticación de un usuario en Firebase Authentication . . . 32
4.8 Cierre de sesión de un usuario en Firebase Authentication . . . 32
4.9 Eliminación de un usuario en Firebase Authentication . . . 32

B.1 JSON de los Eventos . . . 55
B.2 JSON de los Usuarios . . . 56
B.3 JSON de los Mensajes y Respuestas . . . 57
B.4 JSON de las Fotografías . . . 58

Lista de figuras

3.1 Sistemas . . . 11
3.2 Diagrama de Clases . . . 16
3.3 Vistas Principales App . . . 18

4.1 Tabla Resumen Puntos de Función . . . 20
4.2 Diagrama de Gantt: Planificación Temporal del Proyecto . . . 21
4.3 Incremento 1 . . . 24
4.4 Servicios Google Maps y Google Places . . . 29
4.5 Incremento 2 . . . 30
4.6 Invitaciones y Solicitudes . . . 33

5.1 Subida y descarga de Fotografías . . . 36
5.2 Uso Aplicación . . . 37
5.3 Consumo de recursos . . . 37

A.1 Valor en Puntos de Función . . . 50

A.2 Puntos de Función por Archivo de Datos 50
A.3 Puntos de Función por Requisito 51
A.4 Configuración del Sistema 52

C.1 Vista Inicio de Sesión 59
C.2 Vista Registro 60
C.3 Errores registro 61
C.4 Vista Restablecimiento Contraseña 61
C.5 Vista Principal/Mis Eventos 62
C.6 Vista Invitaciones 63
C.7 Creación del Evento 64
C.8 Vista Búsqueda de Eventos 64
C.9 Perfil de Usuario 65
C.10 Confirmación eliminación cuenta 65
C.11 Vista Principal del Evento/Información del Evento 66
C.12 Menú de Opciones del Evento 66
C.13 Confirmación Abandono Evento 67
C.14 Edición del Evento 67
C.15 Gestion Participantes 68
C.16 Enviar nuevas invitaciones 68
C.17 Vista Participantes 69
C.18 Vista Mensajes 69
C.19 Adición Rápida de Mensajes y Respuesta 70
C.20 Vista Detallada de Mensaje 70
C.21 Vista Galería de Fotografías 71
C.22 Vista de Fotografía Detallada 72
C.23 Vista Comentarios Fotografía 72
C.24 Vista Confirmación Subida 73
C.25 Vista Ubicación del Evento 74

1

Introducción

En este capítulo se va a hacer una breve introducción que nos permita poner un marco general a nuestro proyecto, así como explicar la motivación y objetivos del proyecto y del actual documento.

1.1. Motivación

Esta memoria de TFG tiene como objetivo describir de manera detallada el diseño, desarrollo e implementación del Proyecto "Diseño e implementación de una red social de eventos para Android".

El proyecto se centra en desarrollar un producto software viable y con cabida en un mercado cada vez más competitivo como son las redes sociales. Es incuestionable que la expansión e implantación que ha tenido Internet en las dos últimas décadas ha tenido repercusiones en todos los ámbitos de nuestra vida cotidiana. Uno de estos ámbitos sin duda es el ocio y las relaciones interpersonales, y en esto influyen especialmente las Redes Sociales (RRSS) . Las RRSS han tenido un impacto indudable en la manera en la que nos relacionamos con los demás, conocemos gente nueva e, incluso, los medios para encontrar empleo, relaciones sentimentales, etc.

Una posible definición de red social es la que nos ofrece Ros-Martín [1]: "...una plataforma web cuyo objetivo es la creación de comunidades en-línea mediante la representación de las conexiones personales que los usuarios disponen los unos de los otros. Dentro de estos servicios, los usuarios comparten información mediante la utilización de servicios agregados de mensajería personal, micro-blogging, publicación de fotografías, formación de grupos de interés, etcétera." (pág. 554). Sin embargo, esta definición se podría considerar desactualizada, en tanto que la define como "...una plataforma web...", si bien es cierto que las RRSS nacen en el ámbito web, con el desarrollo de la tecnología y especialmente de los dispositivos móviles inteligentes (smartphones), ya no se restringen únicamente a esta plataforma. De hecho, las últimas RRSS que se han establecido son nativas de plataformas móviles como Android y iOS, un ejemplo de esto serían Whatsapp, Instagram o Snapchat.

Independientemente de la plataforma lo que es seguro es que las RRSS son uno de los productos software más utilizados por la gente en su vida cotidiana, y así lo reflejan las estadísticas. Según estadísticas del 2018 en España hay 25,5 millones de usuarios de redes sociales, lo cual supone un

55 % de la población total y un 85 % de los internautas de nuestro país, con un uso diario promedio de 58 minutos por cada red social a la que se pertenece [2].

Esta breve introducción a las redes sociales, su impacto y utilización nos sirve para poner un marco general al proyecto desarrollado, las RRSS son uno de los principales productos software utilizados por los usuarios de Internet y forman parte de la vida diaria de las personas. En la actualidad no hay ninguna red social que centre su actividad en la organización y compartición de mensajes y fotografías para eventos de manera exclusiva. Existen RRSS de mensajería (p.ej. Whatsapp), RRSS que permiten compartir mensajes y fotografías personales (p.ej. Facebook o Instagram), pero ninguna centrada en eventos. El ejemplo más cercano que se puede encontrar en las RRSS instauradas actualmente es la sección de Grupos de Facebook, que permite compartir mensajes, archivos multimedia, etc. entre un grupo seleccionado de usuarios de la aplicación, así como organizar eventos dentro de la aplicación. En el capítulo 2 se llevará a cabo un análisis comparativo más exhaustivo de las principales aplicaciones del mercado que tienen un marco común al de este proyecto.

El proyecto que se describe en esta memoria tiene su motivación y origen en una necesidad personal. En la organización de un evento personal, requería de un sistema que permitiera su organización a través de información básica del mismo (fecha, hora, localización, etc.) así como permitiera a los asistentes del evento compartir mensajes y fotografías. La intención inicial de la aplicación es que el usuario pudiera ver todas las fotografías publicadas por todos los asistentes, publicar las fotografías que hiciera, y que pudiera descargar las que quisiera en su dispositivo, es decir, una red social centrada en eventos. Así como publicar y leer mensajes de los participantes en tiempo real.

En este contexto nace EventSharing una Red Social de Eventos, una app nativa que permite a sus usuarios la organización y gestión de eventos, y en la que los participantes de los eventos pueden compartir mensajes y fotografías. Por un lado, el organizador tiene la posibilidad de compartir toda la información del evento con los invitados (localización, hora, características...), generar expectación a través de mensajes, interactuar en tiempo real con los asistentes o enviar fotografías sobre su preparación. Por otro lado, los invitados tendrán en su móvil toda la información del evento, pueden recibir actualizaciones del organizador y compartir todas las fotos que realicen desde el móvil, así como descargar en su teléfono las que hayan hecho otros invitados.

En cuanto a los usuarios potenciales,es decir, a quién va dirigida esta aplicación, son cualquier persona que posea un smartphone. En la actualidad el desarrollo se centrará en la versión para dispositivos Android, por lo que cualquier persona con un dispositivo Android podría instalar y utilizar la aplicación. Esto hace que el mercado sea muy amplio, ya que como se adelantaba en esta misma introducción únicamente en nuestro país tenemos aproximadamente 25,5 millones de usuarios.

La actual memoria se centra en describir el diseño, el desarrollo y los resultados obtenidos en la implementación de EventSharing para el sistema operativo Android.

1.2. Objetivos

Los objetivos del proyecto se pueden resumir en los siguientes:

O-1.– Crear una aplicación móvil para Android que permita a los usuarios crear y gestionar eventos privados. En la cual los participantes de los eventos puedan compartir mensajes y fotografías con el resto de participantes.

O-2.– El organizador del evento tendrá posibilidad de establecer una serie de parámetros que permitan la organización del evento tales cómo título, descripción, fecha, hora y localización.

O-3.– La aplicación debe ser atractiva e intuitiva para los usuarios. Para ello, el diseño de la aplicación se basará en el diseño actual de las principales redes sociales existentes en el mercado, de manera que para cualquier usuario de redes sociales el entorno le resulte familiar.

O-4.– Cualquier usuario con un dispositivo Android con una versión superior a la API 21, debe poder descargar, instalar y utilizar la aplicación desarrollada.

O-5.– Crear una red social con un mercado potencial significativo y que pueda ser competitiva en el mercado actual de aplicaciones móviles y de las redes sociales.

O-6.– Crear un sistema con un diseño que facilite la futura implementación de nueva funcionalidad, así como su mantenimiento.

O-7.– Desarrollar el sistema y la base de datos sobre la que se apoya el mismo de manera que asegure la escalabilidad del mismo.

1.3. Organización de la memoria

En esta sección se va a explicar la organización y estructura de esta memoria, la cuál se ha realizado de manera que el lector pueda conocer de manera detallada todos los aspectos relacionados con el diseño, desarrollo y los resultados del proyecto. Así, se ha decidido seguir una estructura incremental, de manera que se explicarán tanto el diseño como el desarrollo del proyecto dando una idea general y luego especificando detalladamente cada uno de los aspectos más importantes.

Los capítulos que componen la actual memoria son los siguientes:

- **Capítulo 2: Estado del Arte -** capítulo en el que se realizará un análisis comparativo entre la aplicación descrita en este proyecto y otras aplicaciones tanto móviles como web que tienen un marco de negocio similar al propuesto.
- **Capítulo 3: Diseño -** En este capítulo se pretende especificar el diseño realizado para la aplicación. Se especificarán las funcionalidades de la aplicación, el modelo de clases diseñado y el diseño visual de la aplicación.
- **Capítulo 4: Desarrollo -** En este capítulo se especificará todo el proceso de desarrollo de la aplicación, desde la planificación del mismo hasta la implementación.
- **Capítulo 5: Integración, pruebas y conclusiones -** En este capítulo se realizará un análisis de los resultados obtenidos en la implementación.
- **Capítulo 6: Conclusiones -** En el capítulo final de la memoria se recogerán las conclusiones en relación a los resultados obtenidos y el trabajo futuro a partir del punto de desarrollo donde conclúye este proyecto.

2

Estado del arte

En este capítulo se va a realizar un análisis comparativo entre la aplicación que se define en este proyecto y otras aplicaciones tanto móviles como web del mercado. El objetivo de este capítulo es dar un marco general del mercado competitivo de la aplicación y analizar las bondades y limitaciones de cada una de las aplicaciones similares a EventSharing y cómo se pretende subsanar las limitaciones encontradas con nuestra aplicación.

En primer lugar, se va a realizar un análisis de las aplicaciones más utilizadas en la organización de eventos.

Eventbrite

Eventbrite es una aplicación para la organización de eventos. Permite a los usuarios crear eventos, estableciendo título, descripción, fecha, hora y ubicación del mismo, la aplicación permite al organizador establecer si es un evento público o privado. Los usuarios pueden buscar eventos cercanos, así como la compra de entradas de los mismos. Se ha recalcado Eventbrite ya que es una de las aplicaciones de este ámbito con mayor número de usuarios. Sin embargo, existen otras con funcionalidades similares como Fever, Pro Party Planner o Ticketea Checkpoint.

- **Bondades:**
  - La aplicación permite al organizador añadir mucha información relacionada con la organización del evento.
  - Herramienta útil en la organización de eventos públicos, permitiendo la venta directa de entradas, la obtención de estadísticas con respecto a ventas, marketing, etc.
  - Gestión de invitaciones a los eventos.
  - Promoción de eventos por localización.
  - Aplicación disponible en Android, iOS y Web.
- **Limitaciones:**
  - Aplicaciones independientes para la búsqueda y participación de eventos y para la creación de eventos. Los organizadores tienen que tener una aplicación independiente para la organización, gestión y edición del evento.
  - No existe interacción entre participantes ni entre el organizador y los participantes más allá de la descripción del evento.
  - No existe la posibilidad de compartir fotografías de los eventos entre los participantes.

- **Conclusión:** Aunque estamos ante una muy buena app en lo que respecta a la organización de eventos, el concepto de esta aplicación se aleja de lo que busca ser EventSharing, una red social. En este sentido, esta aplicación tiene una finalidad distinta a la que pretende EventSharing ya que no permite la mensajería entre usuarios ni la compartición de fotografías. Por tanto, está más orientada a la organización de eventos que a ser un medio por el cual los participantes de un evento puedan interactuar, intercambiar y compartir con el resto de los participantes tanto antes, como durante, como después de la realización del evento.

Aplicaciones para tipos de eventos determinados

Existen algunas aplicaciones que permiten organizar y gestionar eventos que están centrados en un único tipo concreto de evento. Un ejemplo de este tipo de aplicaciones son las aplicaciones para la organización de bodas. Muchas de estas aplicaciones además permiten compartir fotografías entre los usuarios. Algunos ejemplos de este tipo de aplicaciones son: Wedshoot ó Webdingsapp.

- **Bondades:**
  - Algunas de estas aplicaciones permiten compartir fotografías entre los participantes del evento. Permitiendo la subida y descarga de fotografías.
  - Están centradas en eventos privados por lo que tienen sistemas de invitaciones y participación al evento similares a los que se buscan para EventSharing.
- **Limitaciones:**
  - Su principal limitación es que se centran en un tipo de evento muy concreto, no pudiendo ser utilizadas para otros tipos de eventos que no sean para el que han sido diseñadas.
  - Ninguna tiene un sistema de mensajería entre los participantes del evento.
  - En general, suelen tener bastantes limitaciones en lo que se refiere a la organización del evento.
- **Conclusión:** Este tipo de aplicaciones tienen un sistema de compartición de fotografías que se acerca a lo que se pretende en el ámbito de las fotografías en EventSharing. Sin embargo, se centran en tipos concretos de eventos mientras que EventSharing pretende ser una aplicación que pueda ser utilizada en cualquier tipo de evento de ocio. Además, no se ha encontrado ninguna que permita la comunicación directa a través de mensajes entre los participantes del evento.

Fuera del ámbito de la organización de eventos encontramos otras aplicaciones con funcionalidades similares a las que se pretenden cubrir con EventSharing.

Facebook

Facebook es una de las principales redes sociales actualmente en el mercado con más de 1,5 billones de usuarios activos diarios. Es una red social que abarca una gran cantidad de ámbitos desde la publicación de fotos y la mensajería hasta, lo que nos ocupa en este documento, la organización y compartición de eventos. Como se adelantó en la introducción, en Facebook es donde encontramos una funcionalidad más similar a la que se propone para esta aplicación. En 2016 Facebook lanzaba los eventos dentro de su red social. Los eventos de Facebook permiten crear un evento, compartiendo la información del mismo como título, descripción, fecha, hora y localización. Además, actualmente permite a los usuarios compartir mensajes y subir fotografías [3].

- **Bondades:**
  - Facilidad en la organización del evento, pudiendo establecer la información relativa al evento.

- Posibilidad de los participantes de publicar mensajes en el tablón del evento.
- Posibilidad de los participantes de subir fotografías al evento para que el resto de participantes las vean.
- Permite al organizador gestionar los participantes, visualizando qué usuarios participan en el evento, invitar nuevos participantes o expulsar participantes.

- **Limitaciones:**
  - No permite la descarga de imágenes ni, por tanto, la descarga múltiple de imágenes desde la aplicación, por lo que permite compartir fotografías pero no puedes descargártelas en tu dispositivo.
  - El sistema de mensajería esta mezclado con otro tipo de publicaciones como por ejemplo, las fotografías, de esta manera los mensajes se pierden en el timeline y pierden importancia en la aplicación.
  - Las opciones de localización que ofrece son reducidas.
- **Conclusión:** A pesar de que es el ejemplo más cercano que se encuentra al concepto de red social que se pretende desarrollar con EventSharing, la funcionalidad que ofrece Facebook en relación a los eventos difiere en ciertos aspectos a la que se pretende desarrollar. En primer lugar, con respecto a la descarga de imágenes, EventSharing pretende facilitar a los participantes la descarga de las imágenes en sus dispositivos. A parte de la visualización en la propia aplicación, se busca que cada participante pueda seleccionar las imágenes que le gusten de un determinado evento y descargarlas en su dispositivo. Además, como se comentaba en las limitaciones la mensajería no tiene una sección propia dentro de los eventos, si no que aparece entremezclada con otro tipo de publicaciones. Fuera del aspecto de la funcionalidad, es un servicio que ofrece Facebook pero no es una aplicación centrada en los evento, sino un servicio adicional dentro de una red social que abarca otros muchos servicios y ámbitos.

Google Maps

Google una de las empresas de servicios informáticos más importantes del mundo ha anunciado este mismo Marzo de 2019 una nueva funcionalidad para su servicio de Google Maps, que permitirá la publicación de eventos públicos en sus mapas. Aunque la información al respecto todavía es limitada y no se sabe cómo implementará estos eventos, la información que ha publicado la empresa parece apuntar a que el usuario podrá añadir eventos al mapa, con información sobre el evento (título, descripción, fecha, hora y localización). así como la publicación de imágenes sobre el evento [4].

Como la información actual sobre esta futura funcionalidad es muy limitada es difícil predecir sus bondades y limitaciones. Por la información que existe actualmente en las redes, parece que no estará tan orientada a ser una red social, sino que estará más enfocada a ser una herramienta de organización de eventos.

3

Diseño

3.1. Funcionalidad

En esta sección se va a describir la funcionalidad que se desea cubrir con la aplicación. Para facilitar la comprensión de dicha funcionalidad, se ha decidido dividir la misma en diferentes sistemas que conformarán el sistema global, así como diferentes subsistemas que conformarán cada uno de los sistemas principales.

3.1.1. Definición de Sistemas

En primer lugar se van a describir todos los sistemas y subsistemas que componen la aplicación, y se va a dar una descripción general de la funcionalidad que englobarán.

Cabe destacar que se ha decidido dividir el sistema global de la aplicación en dos sistemas principales que son el Sistema de Usuarios y el Sistema de Eventos.

Sistema de Usuarios: Este sistema englobará la funcionalidad asociada a la creación y gestión de los perfiles de los usuarios de la aplicación. Es un sistema de gran importancia ya que es el que contiene la funcionalidad que permite el acceso a la aplicación de los usuarios, y por lo tanto su funcionalidad incluye también aspectos relacionados con la seguridad de la aplicación.

Subsistema de Registro: El subsistema de registro contiene toda la funcionalidad asociada al registro de usuarios en la aplicación y la creación de nuevos perfiles. El registro a la aplicación se realizará a través de correo electrónico, pudiendo registrarse cualquier persona con una cuenta de correo electrónico válida.

Subsistema de Inicio de Sesión: El subsistema de inicio de sesión englobará la funcionalidad asociada al logueo en la aplicación y la recuperación de contraseñas perdidas. Cualquier persona que haya completado satisfactoriamente el

proceso de registro y validación de la cuenta podrá acceder a los servicios de la aplicación.

Subsistema de Gestión de Perfiles: El subsistema de gestión de perfiles engloba la funcionalidad asociada a la capacidad de los usuarios de gestionar sus perfiles en la aplicación, así como la eliminación de la cuenta o perfil. Por tanto, este subsistema es el que va a permitir personalizar algunos aspectos del perfil de los usuarios.

Sistema de Eventos: Este sistema engloba la funcionalidad asociada a los eventos. Este sistema es el principal de la aplicación y el que más funcionalidad va a tener asociada. Este sistema permitirá a los usuarios la creación, gestión y participación en los eventos, así como la interacción de los usuarios en los eventos a los que pertenezcan a través de mensajes y la subida de fotografías.

Subsistema de Creación y Gestión de Eventos: El subsistema de creación y gestión de eventos engloba la funcionalidad que permite a los usuarios crear nuevos eventos, la búsqueda de eventos y la gestión y edición de los mismos.

Subsistema de Gestión de Participantes: El subsistema de gestión de participantes engloba la funcionalidad que une usuarios y eventos. En primer lugar, todos los usuarios podrán solicitar el acceso a los eventos de la aplicación, así como ser invitados a eventos y abandonar los eventos de la aplicación. Además, los usuarios podrán consultar la lista de participantes del evento teniendo acceso a los perfiles de dichos usuarios. Por su parte el organizador de un evento determinado podrá invitar a otros participantes al evento, expulsar participantes del evento y aceptar solicitudes de acceso de otros usuarios al evento.

Subsistema de Mensajería: El subsistema de mensajería contiene la funcionalidad asociada a los mensajes entre los participantes de un evento. Todos los usuarios que pertenezcan a un determinado evento podrán publicar mensajes nuevos, así como responder a los mensajes de otros participantes. Además, podrán leer los mensajes publicados por otros participantes del evento.

Subsistema de Fotografías: El subsistema de fotografías engloba la funcionalidad asociada a las fotografías publicadas por los participantes de un evento. Este subsistema es uno de los principales de la aplicación, ya que tal y como se ha especificado en la motivación (1.1) del proyecto, uno de los principales servicios que se pretendía dar a los usuarios es la compartición de fotografías de eventos. Este subsistema engloba la funcionalidad de subida, descarga, eliminación y visualización de las imágenes publicadas por los participantes de un evento. Adicionalmente, los participantes de un evento también podrán publicar comentarios en las fotografías que hayan sido subidas al evento, lo que otorga

a este subsistema una mayor interacción entre los usuarios.

Subsistema de Localización: El subsistema de localización engloba la funcionalidad relacionada con la localización de los eventos en el mapa. Este subsistema permite al organizador establecer una localización en la que se realizará el evento y modificarla. Además, permitirá a los participantes del evento consultar dicha localización en el mapa.

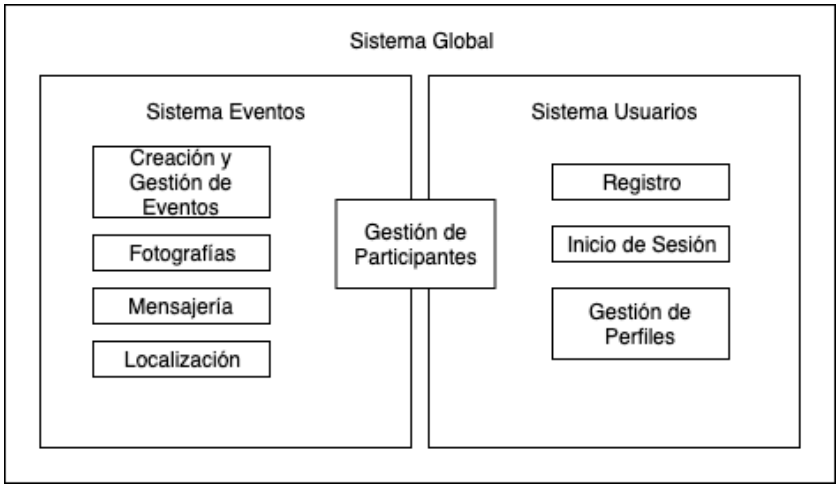


Figura 3.1: Representación visual de los sistemas y subsistemas.

3.1.2. Requisitos Funcionales

En esta sección se van a definir los requisitos funcionales de la aplicación. Para facilitar la organización de los mismos vamos a agrupar los requisitos funcionales en base a los sistemas y subsistemas explicados en la sección 3.1.1.

El objetivo de esta sección es especificar de manera detallada los requisitos funcionales que se desean en la aplicación, es decir, la funcionalidad mínima que nuestro sistema debe cubrir para ser un producto software aceptable.

Sistema de Usuarios

Subsistema de Registro

RF-1.– Registro: El sistema debe permitir a cualquier usuario que posea un correo electrónico válido registrarse en la aplicación.

RF-2.– Correo electrónico: El usuario deberá suministrar un correo electrónico válido. El sistema comprobará que el correo electrónico suministrado sea un correo electrónico válido.

RF-3.– Cuenta única: Únicamente podrá haber una cuenta asociada a un determinado correo electrónico. El sistema comprobará que el correo electrónico introducido por el usuario no esté actualmente asociado a ninguna cuenta de usuario.

RF-4.– Datos de perfil: El sistema registrará algunos datos del usuario como nombre, apellidos y nombre de usuario.

RF-5.– Nombre de usuario: El nombre de usuario deberá ser único en el sistema, en caso de que el usuario introduzca un nombre de usuario no disponible el sistema le alertará. El nombre se usuario deber tener entre 3 y 15 carácteres y no puede tener ni mayúsculas ni espacios. El sistema comprobará que el nombre de usuario introducido no este actualmente en uso por otro usuario.

RF-6.– Contraseña fuerte: El sistema no permitirá el registro con contraseñas débiles. Se considerará que una contraseña es fuerte cuándo su longitud sea de entre 8 y 15 caracteres y con al menos una letra mayúscula y un número. El sistema comprobará la fortaleza de la contraseña introducida por el usuario.

Subsistema de Inicio de Sesión

RF-7.– Validación del correo electrónico: El usuario deberá validar su correo electrónico antes de realizar el primer inicio de sesión en la aplicación, por lo que será un proceso único. Para realizar la validación el sistema enviará un correo electrónico al usuario en el que se le dará un enlace de validación, una vez que el usuario haga click en el enlace, su cuenta quedará validada.

RF-8.– Inicio de sesión: Para iniciar sesión el usuario deberá introducir el correo electrónico con el que realizó el registro y la contraseña indicada en el proceso de registro.

RF-9.– Mantenimiento de la sesión: Una vez realizado el inicio de sesión, la sesión permanecerá abierta aunque el usuario cierre la aplicación.

RF-10.– Recuperación de contraseña: El sistema permitirá al usuario restablecer su contraseña en caso de pérdida u olvido de la misma. Para ello, el usuario podrá solicitar el restablecimiento de su contraseña y el sistema le enviará un correo electrónico con el formulario para restablecer dicha contraseña.

Subsistema de Gestión de Perfiles

RF-11.– Datos inmodificables: El correo electrónico y el nombre de usuario no podrán ser modificados por el usuario.

RF-12.– Datos modificables: El sistema permitirá realizar cambios en el nombre y apellido del usuario.

RF-13.– Imagen de perfil: El usuario podrá modificar su imagen de perfil seleccionando una imagen de la galería de su dispositivo.

RF-14.– Correo electrónico y nombre de usuario: El correo electrónico y el nombre de usuario aparecerán en el perfil del usuario para poder ser consultados.

RF-15.– Cambio de contraseña: El usuario podrá solicitar el cambio de contraseña desde su perfil. Para realizar el cambio de contraseña el usuario deberá introducir su contraseña actual y la nueva contraseña, que deberá ser una contraseña válida según los requisitos espeficicados en el requisito de Contraseña fuerte. En caso de que el usuario no conozca su actual contraseña no podrá realizar el cambio de contraseña desde su perfil, sino que tendrá que cerrar su sesión y realizar el proceso descrito en el requisito de Recuperación de contraseña.

RF-16.– Cierre de sesión: La sesión se cerrará únicamente si el usuario lo solicita a través de un sistema de cierre de sesión disponible desde su perfil de usuario.

RF-17.– Eliminación de la cuenta: El usuario podrá solicitar la eliminación de la cuen-

ta desde su perfil de usuario. La eliminación de la cuenta supondrá la salida del usuario de todos los eventos en los que participe. Una cuenta eliminada no será recuperable, por lo que si el usuario quisiera volver a usar los servicios de la aplicación deberá realizar el proceso de registro de nuevo y obtener una cuenta nueva en la aplicación.

Sistema de Eventos

Subsistema de Creación y Gestión de Eventos

RF-18.– Creacion de eventos: Cualquier usuario logueado podrá crear un evento nuevo. En el proceso de creación del evento, el organizador deberá administrar una serie de datos básicos del evento.

- **RF-18.1.– Título y descripción del evento:** El organizador deberá asignar un título al evento y una descripción del mismo.
- **RF-18.2.– Fecha y hora del evento:** El organizador deberá asignar una fecha y hora de realización del evento.
- **RF-18.3.– Imagen de cabecera del evento:** El organizador asignará una foto de cabecera al evento. Esta imagen podrá ser seleccionada de entre una galería que proporcionará el propio sistema o seleccionada por el usuario desde la galería de su dispositivo.
- **RF-18.4.– Localización del evento:** El organizador deberá seleccionar una localización para el evento. Para ello se le proporcionará un mapa al usuario en el que puede seleccionar la localización. Además, el mapa contará con un buscador que permitirá al usuario buscar la localización por calle, ciudad y puntos de interés.
- **RF-18.5.– Invitaciones a participantes:** El organizador podrá enviar invitaciones de acceso al evento a otros usuarios buscándolos por su nombre de usuario.
- **RF-18.6.– Código de identificación:** El sistema asignará al evento un código de identificación único en el sistema, este código no será en ningún caso modificable.

RF-19.– Búsqueda de eventos: El sistema permitirá buscar eventos existentes en la aplicación a partir de su título, el nombre de usuario del organizador o el código de identificación del evento.

RF-20.– Copia de código de invitación: Cualquier participante del evento podrá copiar al portapapeles de su dispositivo el código de identificación del evento.

RF-21.– Añadir el evento a Google Calendar: Cualquier participante podrá añadir el evento a su cuenta de Google Calendar. Este proceso lanzará la aplicación de Google Calendar y establecerá el título, la descripción, la fecha y la hora del evento en el nuevo evento del calendario de manera automática.

RF-22.– Edición del evento: El organizador podrá modificar toda la información del evento (título, descripción, fecha, hora y localización) en cualquier momento tras la creación del evento.

Subsistema de Gestión de Participantes

RF-23.– Invitaciones: El organizador podrá invitar a cualquier usuario registrado en la aplicación a participar en el evento. Este proceso podrá ser realizado en cualquier

momento después de la creación del evento. Un usuario invitado a un evento podrá aceptar o rechazar la invitación. Una vez que un usuario acepte una invitación a un evento, éste tendrá acceso al evento y su contenido.

RF-24.– Solicitudes de acceso: Cualquier usuario logueado podrá solicitar el acceso a cualquier evento creado. El organizador deberá aceptar la solicitud de acceso enviada por un usuario para que éste tenga acceso al evento y su contenido. En caso de que la solicitud de acceso sea rechazada por el organizador, el usuario no podrá acceder al evento ni a su contenido.

RF-25.– Expulsión de participantes: El organizador podrá expulsar a cualquier participante del evento, el cuál dejará de tener acceso de manera inmediata al evento. El organizador no podrá expulsarse a sí mismo del evento.

RF-26.– Consulta de participantes: Cualquier participante podrá consultar qué otros usuarios participan en el evento y consultar sus perfiles de usuario.

RF-27.– Salir del evento: Cualquier participante de un evento podrá salir del mismo, lo que supondrá que el usuario dejará de tener acceso al contenido de dicho evento. En caso de que el organizador abandoné el evento, éste será eliminado del sistema y ningún participante podrá volver a acceder al contenido del evento.

Subsistema de Mensajería

RF-28.– Adición de mensajes: Cualquier participante del evento podrá añadir nuevos mensajes al evento.

RF-29.– Respuestas a los mensajes: Cualquier participante podrá responder a un mensaje existente en el evento. Las respuestas quedarán asociadas a dicho mensaje.

RF-30.– Consulta de mensajes y respuestas: Cualquier participante podrá consultar los mensajes creados dentro del evento así como las respuestas a los mismos. Además, podrá ver que participante ha escrito cada mensaje o respuesta.

RF-31.– Visualización de la última respuesta: En la vista de mensajes aparecerá la última respuesta del usuario logueado a los mensajes del evento.

Subsistema de Fotografías

RF-32.– Subida de fotografías: Cualquier participante podrá subir nuevas fotografías al evento, las cuáles quedarán añadidas a la galería del evento.

RF-33.– Eliminación de fotografías: Las fotografías podrán ser eliminadas por el usuario que las ha subido al evento o por el organizador del mismo. El organizador podrá eliminar cualquier fotografía de la galería del evento.

RF-34.– Descarga de fotografías: Cualquier participante podrá descargar las fotografías de la galería del evento a su dispositivo. Cuando se realice la descarga se creará un álbum en la galería del dispositivo, en el caso de que no existiera previamente, donde se guardaran todas las fotografías descargadas.

RF-35.– Visualización de la galería: El sistema dispondrá de un modo de visualización general de la galería en el que aparecerán todas las fotografías pertenecientes a la galería del evento.

RF-36.– Visualización detallada: Cada fotografía de la galería podrá ser visualizada de manera detallada (ampliada), y única, es decir, visualizando una única foto en la pantalla. En la visualización detallada también se podrá consultar que usuario ha subido la fotografía que se está observando.

RF-37.– Transición de fotografías en visualización detallada: En el modo de visualización detallada se podrá transitar hacia delante o hacia detrás en la galería de fotografías haciendo swipe a derecha o izquierda. Por lo tanto, los usuarios podrán visualizar la galería completa sin tener que salir del modo de visualización detallada.

RF-38.– Comentarios en las fotografías: Cualquier participante podrá añadir comentarios a las fotografía de la galería del evento y consultar los comentarios de otros participantes en las fotografías.

Subsistema de Localización

RF-39.– Consulta de la localización del evento: Cualquier participante podrá consultar la localización del evento en un mapa. La localización aparecerá claramente marcada en el mapa junto con el título, la fecha y la hora del evento.

RF-40.– Acceso directo a Google Maps: El mapa permitirá el acceso directo a la aplicación Google Maps, permitiendo al usuario abrir la localización del evento o iniciar la navegación hacia la localización en dicha aplicación.

3.1.3. Requisitos No Funcionales

El objetivo de esta subsección es describir de manera detallada los requisitos no funcionales que se han determinado para la aplicación.

RNF-1.– Rapidez de respuesta: La transición entre vistas de la aplicación se hará de forma rápida sin suponer un retardo significativo.

RNF-2.– Carga de imágenes: La aplicación tiene un gran número de imágenes y fotografías, por lo que se espera que la carga de las mismas no tenga tiempos de carga muy elevados. Para ello siempre que sea posible se utilizarán herramientas que provean cacheo de imágenes para la carga de fotografías en la aplicación.

RNF-3.– Consumo de datos: En relación con el anterior punto, las imágenes y fotografías pueden suponer un coste excesivo en el consumo de datos móviles para la carga de dichas imágenes. Para reducir este coste, se hará uso de thumbnail's , de manera que salvo que el usuario solicite la descarga o visualización detallada de una imagen, la descarga automática de imágenes será siempre en su versión reducida.

RNF-4.– Tratamiento de imágenes: Cuando se vaya a realizar la subida de una imagen, además de generar el thumbnail de la imagen descrito en el punto anterior, se reducirá al máximo el tamaño de la imagen sin que se vea afectada la calidad de la misma. Esto permitirá ahorrar espacio de almacenamiento en la nube y permitirá que la descarga de imágenes sea menos costosa en consumo de datos y más rápida.

RNF-5.– Adaptación a diferentes resoluciones de pantalla: La aplicación se desarrollará asegurando su correcta adaptabilidad a diferentes resoluciones de pantalla. De esta manera debe asegurarse la correcta visualización tanto en pantallas de móviles como en pantallas de tablet.

RNF-6.– API mínima de instalación: Seleccionar una API que asegure el mayor número de usuarios sin perder funcionalidad.

RNF-7.– Seguridad y Privacidad: Asegurar la seguridad y privacidad de los usuarios y los datos de los mismos en la aplicación. Así mismo, es necesario que el sistema de autenticación sea seguro en relación al almacenamiento y gestión de los datos de autenticación (correo electrónico y contraseña).

3.2. Modelo

3.2.1. Diseño de clases

En esta sección se va a describir el diseño de clases para la aplicación, así como sus relaciones.

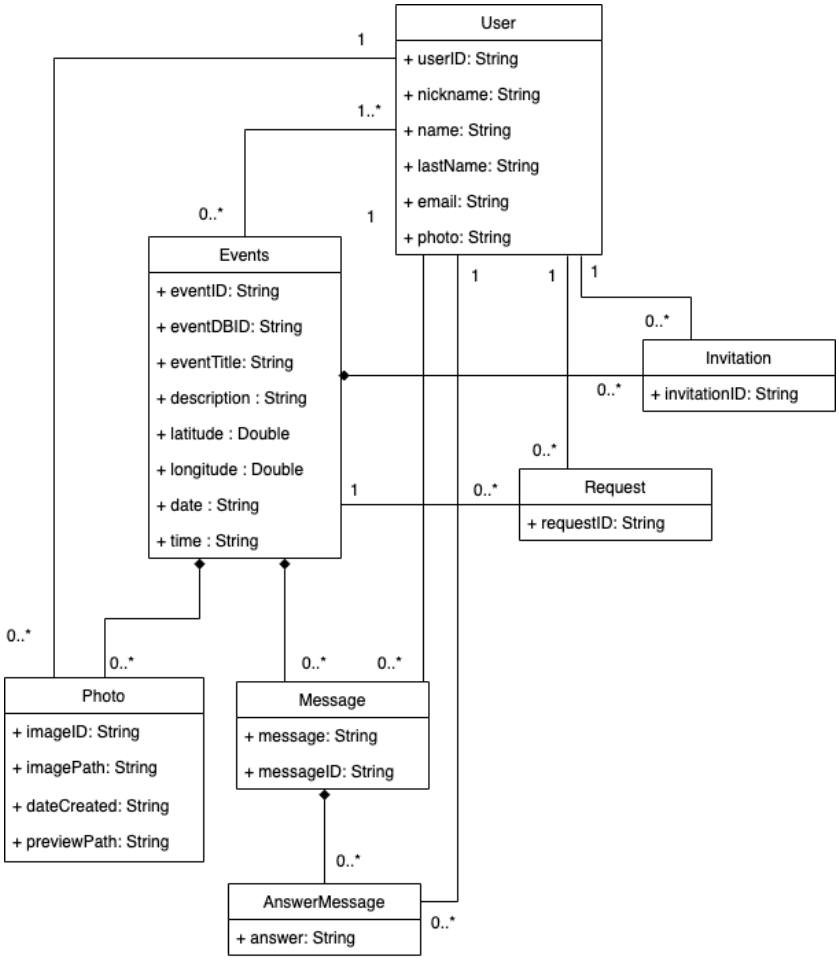


Figura 3.2: Diagrama de clases

Como se puede apreciar en la figura 3.2 en la aplicación encontramos las siguientes clases:

User: Contiene toda la información del usuario logueado en ese momento en la aplicación. Es una de las clases principales de la aplicación, como podemos observar todas las demás clases que conforman la aplicación tienen algún tipo de relación con la clase *User*.

Event: La clase Event contiene toda la información relacionada con un evento. En cuanto a sus campos cabe destacar que tiene dos campos de *ID*, *eventID* y *eventDBID*, el primero de ellos es el identificador del evento dentro de la aplicación, el segundo es el identificador del evento asignado en la base de datos. En cuanto a los campos *latitude* y *longitude* determinan las coordenadas en la que se realizará el evento y permiten cargar el mapa en la aplicación.

Como podemos ver esta clase tiene dos relaciones de composición con las clases *Photo* y *Messages*. Esto se debe a que ambas clases forman parte del evento, y en caso de eliminar el evento se eliminarían todas las fotos y todos los mensajes del evento. Además, se relaciona con la clase *User*, de manera que un usuario puede pertenecer a múltiples eventos, y un evento puede tener múltiples usuarios.

Photo: Esta clase define una fotografía de un evento, en cuanto a sus campos caben destacar dos campos que son el *imagePath* y el *previewPath*, ambos son URL's que permiten cargar la foto desde Firebase Storage a la aplicación.

Message: Clase que define los mensajes en el tablón de mensajes de un evento. Como podemos observar en el diagrama, los mensajes tienen una relación de composición con la clase *AnswerMessage*, de manera que un mensaje puede tener múltiples respuestas y estas existen únicamente en relación al mensaje que componen.

AnswerMessage: La clase *AnswerMessage* define respuestas a los mensajes.

Invitation: Esta clase define las invitaciones de usuarios a un evento. Las invitaciones relacionan usuarios y eventos, de manera que el organizador de un evento puede enviar invitaciones a los usuarios que quiere que formen parte del evento. De esta manera *Invitation* se relaciona tanto con *Event* como con *User*.

Request: Esta clase define las solicitudes de acceso a un evento. Son similares a las invitaciones, salvo por que en este caso son enviadas por el usuario a un evento determinado para solicitar ser admitido en dicho evento. Al igual que la clase *Invitation*, se relaciona con la clase *Event* y la clase *User*.

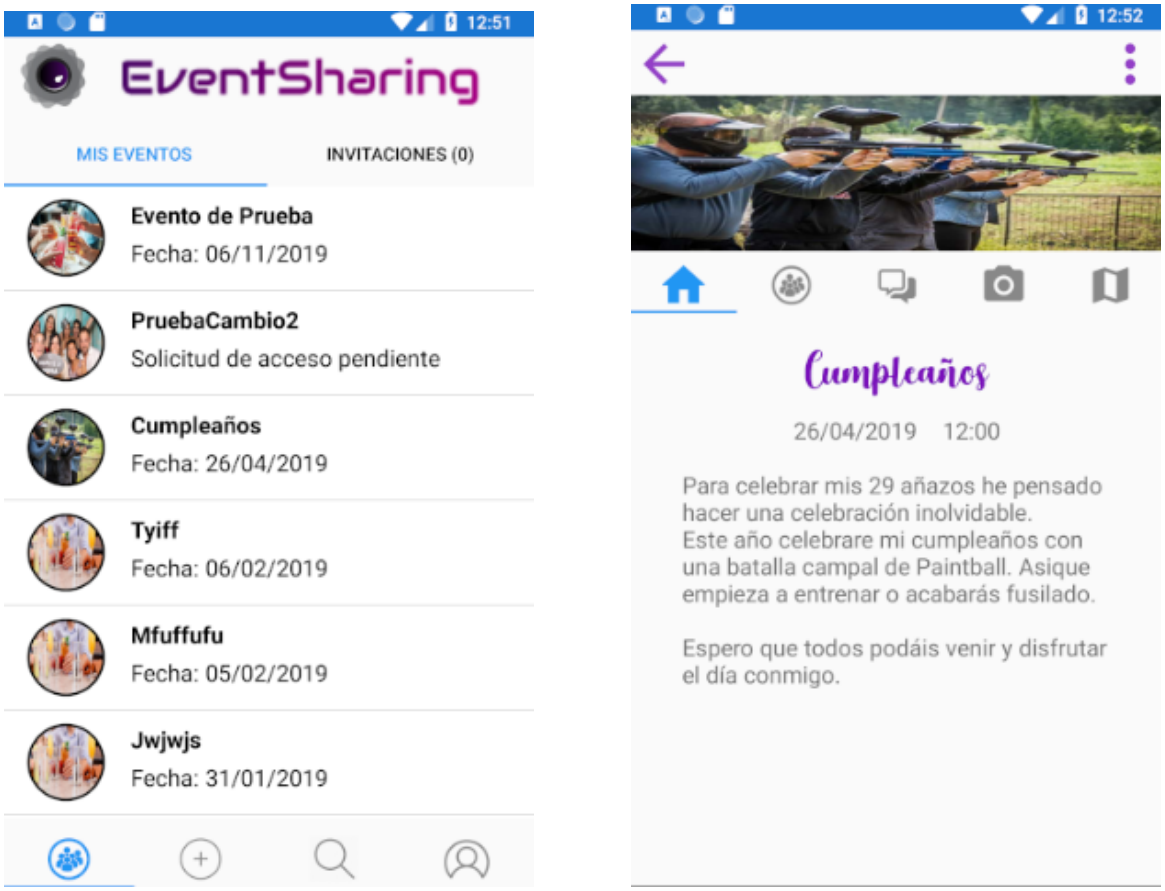
3.3. Diseño Visual y Conceptual

En esta sección se va a detallar el diseño visual y conceptual de la aplicación. Cabe destacar, que debido a que el introducir capturas de todas las vistas de la aplicación requeriría de mucho espacio de esta memoria, se ha optado por presentar sólo las dos vistas principales en esta sección. Sin embargo, se anima al lector a consultar el apéndice C de esta memoria, en el que podrá ver el manual de usuario con capturas de todas las vistas de la aplicación con detalle.

El principal objetivo con respecto al diseño visual de la aplicación ha sido que el usuario se sintiera cómodo y que el sistema le pareciera familiar y similar al que se encuentra en otras aplicaciones de redes sociales. Por ello, se ha buscado que la manera de presentar ciertos aspectos como los mensajes o fotografías no difiera significativamente con respecto a otras aplicaciones ampliamente asentados en el mercado de redes sociales.

Así mismo, se decidió presentar toda la funcionalidad en dos vistas principales que varíen su contenido en función de pestañas. Esto hace que no haya excesivas transiciones entre vistas y da un

aspecto más profesional a la aplicación.



(a) Vista Principal App

(b) Vista Principal Evento

Figura 3.3: En la Figura 3.3(a) podemos ver la vista principal de la aplicación, una vez el usuario se ha logueado. En la Figura 3.3(b) podemos ver la vista principal de un evento.

La primera de estas vistas es la vista principal de la aplicación (figura 3.3(a)), desde la que los usuarios logueados tendrían acceso a los eventos a los que pertenecen, la creación de eventos, la búsqueda de eventos y la gestión de su perfil de usuario.

La segunda (figura 3.3(b)), es la vista principal del evento desde donde tanto organizador como participantes pueden acceder a todo el contenido y funcionalidad de la aplicación, los participantes, mensajes, fotografías y mapa del evento. Cabe destacar que en la barra de acción superior, el usuario puede volver a la vista principal o ,pulsando sobre el icono de puntos, se despliega un menú donde puede acceder a diferentes funcionalidades y configuraciones. En este menú, el organizador tendrá las herramientas de gestión del evento.

4

Desarrollo

En este capítulo se va a describir el proceso de desarrollo del proyecto, desde la planificación hasta la codificación e implementación del mismo. Tal y como veremos en el apartado de Planificación (4.1) de este capítulo, el desarrollo se ha dividido en dos etapas o incrementos. Por ello, se ha decidido organizar este capítulo en cuatro secciones: Planificación (4.1), Aspectos Generales (4.2), Desarrollo de la Beta (4.3) y Desarrollo del Producto Final (4.4)

En primer lugar se va a explicar de manera detallada la planificación llevada a cabo para el desarrollo de la aplicación.

4.1. Planificación

A continuación, se va a detallar la planificación temporal del proyecto. El desarrollo del proyecto se llevará a cabo utilizando un ciclo de vida incremental iterativo, dividido en 2 incrementos y una fase final en la que se realizará la revisión y la documentación final del proyecto.

Para establecer la planificación temporal de manera más exacta se ha decidido utilizar el método de estimación por Puntos de Función IFPUG-FPA [5], lo cual nos dará una estimación más exacta tanto de la complejidad del sistema como del coste temporal de su desarrollo.

Para ello en primer lugar, debemos establecer la complejidad de cada subsistema, y por tanto de cada uno de los requisitos funcionales que engloban. En el apéndice A, podemos ver en detalle el cálculo de los puntos de función realizado para este proyecto.

En la figura 4.1 se muestra una tabla resumen de los cálculos realizados en la cual podemos observar los puntos de función desajustados, los puntos de función ajustados y la predicción del número de semanas de desarrollo de cada subsistema.

Componente	Puntos de Función Desajustados	Puntos de Función Ajustados	Semanas de desarrollo
Subsistema de Registro	22	22,88	3
Subsistema de Inicio de Sesion	14	14,56	2
Subsistema de Gestión de Perfiles	29	30,16	4
Subsistema de Creación y Gestión de Eventos	22	22,88	3
Subsistema de Gestión de Participantes	18	18,72	2
Subsistema de Mensajería	12	12,48	1,5
Subsistema de Fotografías	26	27,04	3,5
Subsistema de Localización	7	7,28	1
Funciones y Archivos de Datos	51	53,04	6
Total	**201**	**209,04**	**26**

Figura 4.1: Tabla en la que se muestran los puntos de función desajustados, los puntos de función ajustados y las semanas de desarrollo calculados a través del sistema IFPUG-FPA

En total el sistema tiene 209,04 Puntos de Función. Estos cálculos nos van a permitir obtener un cálculo mucho más aproximado del tiempo necesario para desarrollar cada subsistema y, por tanto, para el desarrollo del sistema global.

Tal y como podemos observar en la figura 4.1 se ha realizado una predicción de las semanas de desarrollo que van a ser necesarias para cada subsistema y para los archivos y funciones de datos, es decir, para el desarrollo e integración de la base de datos del sistema. Para calcular la estimación de semanas de desarrollo se ha establecido que se pueden desarrollar 8 puntos de función a la semana. Teniendo en cuenta esto, obtenemos que el desarrollo del sistema global nos llevará un total de 26 semanas de desarrollo aproximadamente.

Como se ha mencionado al comienzo de la sección, el desarrollo se realizará utilizando un ciclo de vida incremental iterativo, dividido en dos incrementos o etapas y una fase final:

- **Incremento 1 (Fase 1/Beta):** En este primer incremento se desarrollará una beta funcional implementando parte de los subsistemas. Para la beta se ha decidido desarrollar algunos de los subsistemas del Sistema de Eventos, en concreto se desarrollarán los siguientes: Subsistema de Creación y Gestión de Eventos, Subsistema de Mensajería, Subsistema de Fotografías y el Subsistema de Localización. Además se desarrollará una parte de la base de datos del sistema y de las funciones de datos que utilizaremos en el sistema. En total se estima una duración del desarrollo de esta fase de 12 semanas, es decir, un 45 % del proyecto. Al finalizar esta fase se realizará la revisión y documentación necesaria y una prueba de la Beta en un evento real.

- **Incremento 2 (Fase 2):** En el segundo incremento se desarrollará la funcionalidad asociada a los usuarios, desarrollando al completo el Sistema de Usuarios y el Subsistema de Gestión de Participantes, además del diseño de la parte de la base de datos encargada de los usuarios y sus funciones. Al finalizar el desarrollo se tendrá el Sistema Global implementado, con toda la funcionalidad definida en el apartado 3.1. Se estima que esta fase tendrá una duración de 14 semanas, completando las 26 semanas de desarrollo estimadas.
- **Fase Final:** En la fase final se realizará la revisión del producto final y la documentación final del producto software.

Con estos datos podemos realizar un Diagrama de Gantt que nos permita organizar visualmente la planificación temporal del proceso de desarrollo (figura 4.2).

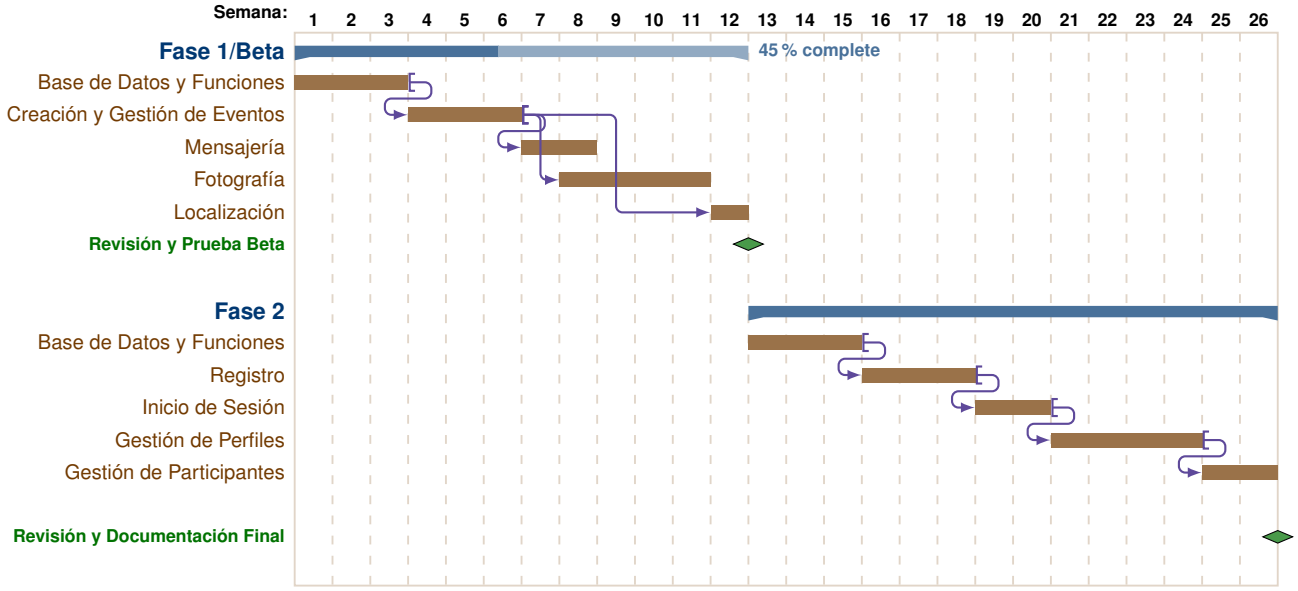


Figura 4.2: Diagrama de Gantt en el que se especifica la planificación temporal del desarrollo del proyecto.

Cabe destacar que la fase final de Revisión y Documentación se realiza a partir de la semana 26 e incluye el desarrollo de este mismo documento. Sin embargo, con el objetivo de simplificar en el diagrama se marca únicamente el inicio de esta fase.

4.2. Aspectos Generales

En esta sección se van a explicar algunos aspectos generales de importancia para el desarrollo de la aplicación.

En primer lugar, el entorno de desarrollo en el que se ha implementado la aplicación es Android Studio. La elección de este entorno de desarrollo se debe a que se trata de un entorno específica-

mente diseñado para el desarrollo de aplicaciones en dispositivos Android. Tiene herramientas muy útiles como un editor de vistas que permite la previsualización de las vistas de la aplicación a medida que se van implementando, un entorno de codificación inteligente que ofrece atajos y sugerencias de funciones a medida que se va programando, un emulador de dispositivos para probar el desarrollo de manera sencilla en distintos dispositivos, además de herramientas de compilación y testeo de la aplicación de mucha utilidad [6].

En cuanto al lenguaje de programación, la aplicación se ha codificado íntegramente en Java. Java, es uno de los principales lenguajes de programación utilizados en la implementación de aplicaciones Android hasta el momento.

Tal y como se adelantaba en los requisitos no funcionales (apartado 3.1.3) uno de los aspectos más importantes a la hora de comenzar el desarrollo de una aplicación Android es elegir el API mínimo que exigirá la aplicación para la instalación en un dispositivo. Este es un aspecto no trivial ya que hay que elegir un API mínimo que asegure que una gran mayoría de los usuarios con dispositivos Android puedan instalar la aplicación, pero no uno tan bajo que afecte a las funcionalidades que pueden ser implementadas al no estar disponibles en API's muy antiguas. En este caso se ha seleccionado la API 21, ya que permite que el 85 % de los usuarios Android instalen la aplicación y no supone una pérdida de funcionalidades significativa.

Para este proyecto se ha decidido utilizar el servicio Firebase de Google para implementar una parte importante de lógica asociada al backend. Firebase utiliza la infraestructura de Google para proveer a los desarrolladores de aplicaciones una serie de herramientas como Bases de Datos en tiempo real, Autenticación de Usuarios mediante diferentes métodos (correo electrónico, cuentas google, etc.), almacenamiento en la nube para multimedia y archivos, funciones automáticas ante eventos en la base de datos, etc. Por tanto, se trata de un servicio muy completo para el desarrollo de ciertas funcionalidades del backend de la aplicación [7]. Además, es una herramienta muy útil en lo que a la escalabilidad de la aplicación se refiere ya que te asegura una adaptación perfecta de los servicios que provee en función del número de solicitudes y consultas que recibe de tu aplicación, lo que asegura la escalabilidad y sin que el servicio se vea interrumpido por falta de recursos en los servidores.

En este proyecto en concreto, se ha decidido utilizar Firebase en la implementación de:

- **Autenticación:** Firebase aporta al programador un sistema fiable, eficiente y seguro para gestionar la autenticación de la aplicación. En este sentido, se delega en Firebase uno de los aspectos más importantes de la seguridad de una aplicación que es el almacenamiento y recuperación de contraseñas asociadas a las cuentas de usuario. Además, provee de funciones que permite la verificación de correos electrónicos y recuperación y cambio de contraseñas a través del envío de correos electrónicos al usuario.
- **Base de Datos:** Se ha decidido implementar la base de datos del sistema en el servicio Firebase. Además del aspecto de seguridad al estar protegida por la infraestructura de Google, Firebase aporta al desarrollador una base de datos que se actualiza en tiempo real y permite dar servicios como mensajería instantánea, actualización de imágenes en el momento, etc. Adicionalmente, la base de datos se basa en objetos json, los cuáles aportan ciertas ventajas ya que son más fáciles de cargar y leer en la aplicación, al aportar una estructura tipo hashmap

muy útil en la recuperación de datos.

- **Almacenamiento en la nube:** El servicio de almacenamiento en la nube de Firebase es una herramienta de gran utilidad para la parte de subida y recuperación de imágenes y fotografías de la aplicación. Al subir una imagen a la nube de Firebase éste, de manera automática, le asocia a la imagen una URL que permite recuperar de manera sencilla la imagen en tu aplicación.

4.2.1. Base de datos

En este subapartado se va a explicar los aspectos generales de la Base de Datos (DB) creada para almacenar la información relativa a los eventos, usuarios, los mensajes y fotografías. Si el lector quiere conocer los aspectos concretos de la base de datos implementada (Objetos JSON implementados, estructura de la base de datos y campos almacenados) puede acudir al apéndice B.

El diseño de la DB y las funciones que se utilizan para recuperar información de la misma o subir información a la DB en la nube, se han ido desarrollando a lo largo de los dos incrementos, por lo tanto, habrá campos en la DB de eventos, mensajes y fotografías que se han añadido durante el segundo incremento.

En primer lugar, se van a discutir las ventajas y desventajas de utilizar una DB no relacional en contraposición a una DB relacional y se va a explicar el porqué de haber elegido este tipo de base datos en nuestro proyecto:

Las principales ventajas de que utilizar una DB no relacional basada en objetos JSON son:

- Mayor facilidad de implementación
- Flexibilidad y adaptación a las necesidades del proyecto.
- Facilidad a la hora de recuperar los datos y trabajar con los resultados de las consultas en objetos de tipo hashmap.
- Adaptabilidad a los cambios. Por ejemplo, la adaptación a la adición de nuevos campos en un objeto sin que afecte al resto de información.
- Estructura jerárquica de los datos.

Por su parte también encontramos ciertas desventajas sobre todo por la potencia de algunos aspectos de las bases de datos relacionales:

- Perdida de la Atomicidad. Uno de los aspectos más potentes de las bases de datos es que si hay que modificar varios campos en varias tablas y falla una de esas actualizaciones, el sistema es capaz de volver automáticamente al estado anterior a través del rollback.
- Consultas anidadas y combinación de tablas. Otro de los aspectos importantes es la capacidad de las bases de datos relacionales de mezclar tablas a través de operaciones join y hacer consultas anidadas para obtener datos más concretos.

Teniendo estas ventajas y desventajas podemos ver más claramente por qué se ha optado por la utilización de una DB no relacional. La razón de mayor peso de las nombradas, es la facilidad para la adaptación a cambios del sistema. Como se ha comentado, en el primer incremento se va a

desarrollar una parte de la DB y en el segundo incremento se implementará la parte restante, esto hace que necesitemos una DB que al cambiar su estructura, añadirle nuevos campos, etc., no se vea afectada en términos de rendimiento. Esto unido a la facilidad que nos brinda en el mantenimiento de este tipo de bases de datos, ya que puedes subir una instancia de una clase Java por ejemplo una instancia de un Evento, y automáticamente te la transforma en un objeto JSON que queda almacenado con todos sus campos, sin tener que hacer un "parseo" manual de los objetos y sus campos. Por ello, se ha optado por utilizar este tipo de DB y no una DB relacional.

4.3. Desarrollo de la Beta

En esta sección se va a explicar el desarrollo realizado para la versión beta de la aplicación. Como ya se especificó en el apartado 4.1, el desarrollo se divide en dos incrementos, el primero de los cuáles se corresponde con el desarrollo de la versión Beta de la aplicación.

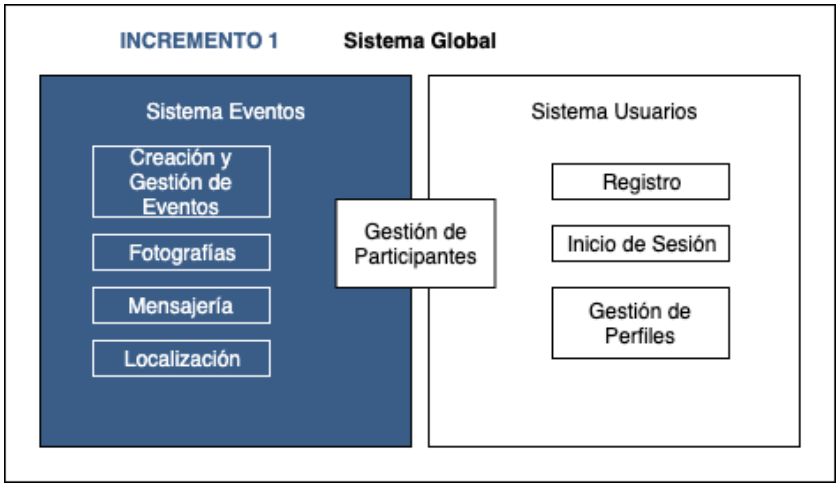


Figura 4.3: Especificación visual de los subsistemas a implementar en el primer incremento.

Tal y como podemos observar en la figura 4.3, en este incremento se implementan los siguientes subsistemas:

- Subsistema de Creación y Gestión de Eventos
- Subsistema de Mensajería
- Subsistema de Fotografías
- Subsistema de Localización

El objetivo principal de este incremento es poder crear un evento y la funcionalidad principal asociada a los eventos de manera que se pudiera hacer una prueba de la aplicación en un evento real y de esta forma hacer un estudio de la viabilidad comercial del producto software, un test de campo del producto y recoger la experiencia de usuarios reales para implementar mejoras y subsanar posibles errores.

Cabe destacar que en este incremento no existen usuarios en la aplicación, por lo que todo usuario con la aplicación instalada tiene acceso al evento y su contenido. Además, al no tener usuarios fue necesario adaptar ciertos aspectos de las funcionalidades y requisitos definidos.

Se van a explicar los aspectos más relevantes del desarrollo de estos subsistemas, no se explicará detalladamente el desarrollo de cada uno de los subsistemas debido a la extensión que requeriría, pero sí que se pretende dar una visión de los aspectos fundamentales del desarrollo de este incremento.

4.3.1. Desarrollo del Subsistema de Creación y Gestión de Eventos

Para la creación de eventos se ha implementado una serie de vistas que van solicitando al usuario toda la información necesaria para crear el evento.

Como se buscaba que este proceso fuera accesible desde la vista principal de la aplicación, se ha utilizado un sistema de fragmentos. Esto nos permite subdividir el proceso en múltiples vistas y presentarlas todas dentro de la misma interfaz haciendo que se vayan sustituyendo unas a otras.

Código 4.1: Ejemplo de transición de diferentes Fragmentos dentro de la misma Actividad

```
1 Fragment fragment = new AddEventFragment();
2 FragmentTransaction transaction = getFragmentManager().beginTransaction();
3 transaction.replace(R.id.add_event_frame, fragment);
4 transaction.commit();
```

En el código se puede ver el sistema de transición entre fragmentos. Lo que se hace es, a través de la instancia del objeto de tipo FragmentTransaction, sustituir la vista que se esta presentando actualmente en el contenedor add_event_frame , de manera que se cambie la vista que se presenta al usuario. De esta manera, si quisieramos cambiar a la vista que recoge la ubicación del evento y cargarla en el mismo contendor sería tan sencillo como hacer lo siguiente:

Código 4.2: Ejemplo de transición de diferentes Fragmentos dentro de la misma Actividad

```
5 transaction.replace(R.id.add_event_frame, new AddEventMapFragment()).commit();
```

Esto es una forma muy potente que nos ofrece Android de poder sustituir elementos concretos de la interfaz sin tener que cambiar la vista general en la que se encuentran dichos elementos.

Esto se ha utilizado a su vez para presentar la interfaz dividida en diferentes secciones con pestañas. Como podemos observar en el diseño de la aplicación, tanto la vista principal de la aplicación como la vista principal de los eventos esta subdividida en diferentes pestañas entre las cuales el usuario puede navegar para poder acceder a todo el contenido de las vistas. La lógica detrás de este

comportamiento es muy similar a la explicada para la transición de fragmentos, con la salvedad de que, para este tipo de vistas divididas en pestañas, Android tiene un tipo de vista especial llamada ViewPager. Esta vista es muy similar a lo que hemos explicado en la transición de fragmentos ya que en definitiva es un contenedor de fragmentos que se controla a través de pestañas, que son las que marcan qué fragmento debe presentársele al usuario. La principal diferencia es que este objeto requiere de un tipo especial de adaptador que extienda la clase FragmentStatePagerAdapter.En definitiva la lógica que debemos implementar en este adaptador es qué fragmento debe cargarse en función de la pestaña que se haya pulsado.

Para la presentación de los diferentes eventos a los que se tiene acceso se ha optado por presentarlos en forma de lista a través de un RecyclerView, esta vista es una herramienta fundamental en el desarrollo Android ya que permite crear listas y cuadrículas y, gracias a objetos de tipo ViewHolder, ahorrar en el espacio de RAM que ocupan estas listas ya que lo que hacen es crear el número de celdas o filas que caben en la vista y después sustituir el contenido de las celdas ya creadas al hacer scroll sobre la lista, de manera que solo se crean las celdas que realmente caben en la vista. Este tipo de vistas requieren de un adaptador que les indique el contenido que debe cargar en cada celda definiendo el ViewHolder para ese RecyclerView, así como definir la lógica de la lista como, por ejemplo, que acción realizar en caso de que se pulse sobre un ítem.

4.3.2. Desarrollo de los Subsistemas de Mensajería y Fotografías

Estos dos subsistemas se ha decidido unificarlos en la explicación ya que tienen bastantes características en común.

Las vistas principales de ambos subsistemas están formadas por un RecyclerView que muestra los mensajes y las fotografías, en una lista y una cuadrícula respectivamente. Ante la pulsación de un elemento de la lista o cuadrícula ambas cambian de actividad, yendo a la vista de vista detallada del mensaje o a la vista detallada de la fotografía.

Del Subsistema de mensajes cabe destacar que para la versión Beta se tuvieron que realizar modificaciones sobre el diseño. La razón principal de esto fue que en este incremento no existe el Sistema de Usuarios, por lo que se optó por añadir un campo adicional a la hora de añadir un mensaje en el que el usuario introducía manualmente el nombre del autor del mensaje. En la vista de detallada del mensaje, que se abre al pulsar sobre un mensaje en la lista de mensajes, se muestra el mensaje y todas las respuestas del mismo en forma de lista.

En cuanto al subsistema de Fotografías, como ya se ha comentado con anterioridad uno de los retos de este subsistema era hacer que la descarga de imágenes fuera eficiente tanto en tiempo de carga de la imagen como en el tamaño de la imagen. Si las imágenes se suben al almacenamiento en la nube tal cual las introduce el usuario en la aplicación, el tamaño de éstas es muy elevado superando en

su mayoría los 2 MB, algunas incluso más. El tamaño de las imágenes es un tema de vital importancia ya que supongamos que un evento tiene 200 imágenes en su galería, si no se hace ningún tratamiento de las mismas, cada vez que el usuario accediera a la galería del evento descargaría casi 0.5 GB de datos. Si el usuario está utilizando datos móviles, con unos cuantos accesos a la galería del evento consumiría la totalidad de los datos de su tarifa móvil. Esto es insostenible, por lo que se optó por realizar los siguientes tratamientos a las imágenes:

- **Thumbnails:** Como ya se ha comentado en el apartado 3.1.3 la primera medida que se ha tomado para evitar el consumo excesivo de datos ha sido la creación de thumbnail's cuando se va a realizar la subida de una imagen. Es decir, crear una versión de tamaño muy reducido que se pueda cargar como previsualización de la imagen en contenedores pequeños como, por ejemplo, la cuadrícula de previsualización de la galería. De esta manera las imágenes con tamaño más grande solo se descargarán en caso de que el usuario solicite la descarga o entre en modo de visualización detallada de la imagen.
- **Reducción de Fotografías:** De la misma manera, vamos a reducir el tamaño que ocupa de la imagen comprimiendo la imagen original pero sin que afecte a la calidad de la imagen.

A continuación, vamos a ver el código que permite la creación de los thumbnail's. Como podemos observar en el código lo que hace la función en establecer el nuevo ancho de la foto, calcular la altura de la foto manteniendo la proporción anchura-altura de la imagen original. Si la imagen fuera menor que el ancho definido, lo cual no debería ocurrir en un caso de una fotografía normal, se va a generar un thumbnail cuadrado con el mismo ancho y alto.

Código 4.3: Función que permite crear un thumbnail a partir de una imagen

```
1  private Bitmap getPreview (Bitmap galleryPic){
2      int previewWH = 300;
3      int origWidth = galleryPic.getWidth();
4      int origHeight = galleryPic.getHeight();
5      int destHeight;
6      if(origWidth > previewWH) {
7          destHeight= origHeight / (origWidth / previewWH);
8      }else{
9          destHeight=previewWH;
10     }
11     return Bitmap.createScaledBitmap(galleryPic, previewWH, destHeight, true);
12 }
```

En cuanto a la reducción del tamaño de las fotografías comprimiendo la imagen original, fue necesario hacer diferentes pruebas para encontrar el grado de compresión que no afectará a la calidad de la imagen en tamaño completo. A continuación, podemos ver el código que realiza la compresión de las imágenes y que es utilizado tanto para la imagen original como para el thumbnail.

Para ejemplificar, el resultado se van a proporcionar datos reales de un tratamiento de una imagen.

- **Tamaño original:** 2,57 MB.
- **Tamaño Imagen Completa Reducido:** 527.740 B.

Código 4.4: Función que comprimir una imagen reduciendo su tamaño en Bytes

```
1 public static byte[] getBytesFromBitmap(Bitmap bm, int quality){
2     ByteArrayOutputStream stream = new ByteArrayOutputStream();
3     bm.compress(Bitmap.CompressFormat.JPEG, quality, stream);
4     return stream.toByteArray();
5 }
```

- **Tamaño del Thumbnail:** 8.324 B

Como podemos ver la reducción del tamaño es más que considerable, la imagen de tamaño completo se reduce un 80 % y el del thumbnail un 97 % con respecto al tamaño original, esto sin suponer una perdida en la calidad de visualización de la imagen.

4.3.3. Desarrollo del Subsistema de Localización

Para el desarrollo del subsistema de localización se ha hecho uso de dos servicios de Google muy potentes que son el servicio de Google Maps para la representación de mapas en la aplicación y el servicio de Google Places para la búsqueda de direcciones y lugares de interés.

Estos servicios permiten ofrecer en la aplicación una funcionalidad muy potente de localización permitiendo al organizador seleccionar la ubicación del evento en un mapa y presentando esta ubicación a los participantes de manera visual.

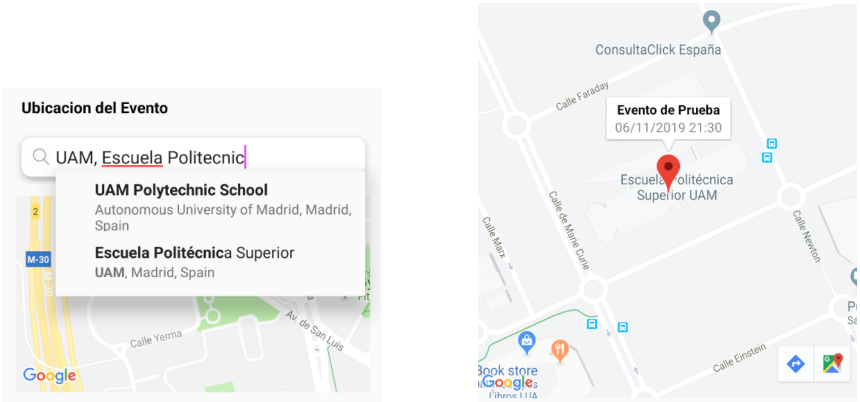
La utilización de Google Places permite presentar al usuario una barra de búsqueda (4.4(a)) que, a medida que el usuario va escribiendo en ella, va presentando diferentes direcciones y/o lugares de interés que el usuario puede seleccionar para autocompletar la búsqueda. Para implementar esta funcionalidad se ha utilizado un adaptador especial, PlaceAutocompleteAdapter, que tiene licencia de Google y que se encarga de buscar en la DB de Google Places para encontrar resultados que coincidan con el texto introducido por el usuario [8].

En cuanto a Google Maps, permite añadir a la aplicación mapas para mostrar ubicaciones de manera exacta a partir de unas coordenadas. Tal y como hemos podido ver en el apartado 4.2.1, el evento guarda la ubicación a partir de la longitud y la latitud, estos valores son utilizados para pintar el marcador que podemos ver en la figura 4.4(b).

Código 4.5: Fragmento de código en el que se muestra cómo se añaden los marcadores al mapa que podemos observar en la figura 4.4(b)

```
1  LatLng position = new LatLng(event.getLatitude(),event.getLongitude());
2  String snippet = event.getDate()+"␣"+event.getTime();
3  MarkerOptions markerOptions = new MarkerOptions();
4  markerOptions.position(position).title(event.getEventTitle()).snippet(snippet);
5  siteMarker = this.googleMap.addMarker(markerOptions);
6
7  this.googleMap.setMinZoomPreference(10);
8  CameraPosition camera = CameraPosition.builder()
9          .target(siteMarker.getPosition())
10         .zoom(17)
11         .build();
12 this.googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(camera));
```

Como podemos observar en el código se crea un objeto de tipo LatLng a partir de la latitud y longitud almacenada en la instancia del evento. Después se crea un objeto de tipo MarkerOptions que será nuestro nuevo marcador en el mapa y al que se le configura con la posición creada, se le pone el título del evento como título del marcador y la fecha y la hora del evento como subtitulo. Después, se añade el mapa el marcador y se hace que la visualización del mapa apunte al nuevo marcador que hemos añadido a través del objeto CameraPosition.



(a) Google Places

(b) Google Maps

Figura 4.4: En la Figura 4.4(a) podemos ver la barra de Búsqueda en el mapa implementada usando Google Places. En la Figura 4.4(b) podemos ver un mapa de la aplicación mostrando la ubicación de un Evento y su información

4.4. Desarrollo del Sistema Final

En esta sección se va a describir el desarrollo del Incremento 2, tal y cómo se especificó en el apartado 4.1 consistirá en la implementación de los siguientes subsistemas:

- Subsistema de Registro
- Subsistema de Inicio de Sesión
- Subsistema de Gestión de Perfiles
- Subsistema de Gestión de Participantes

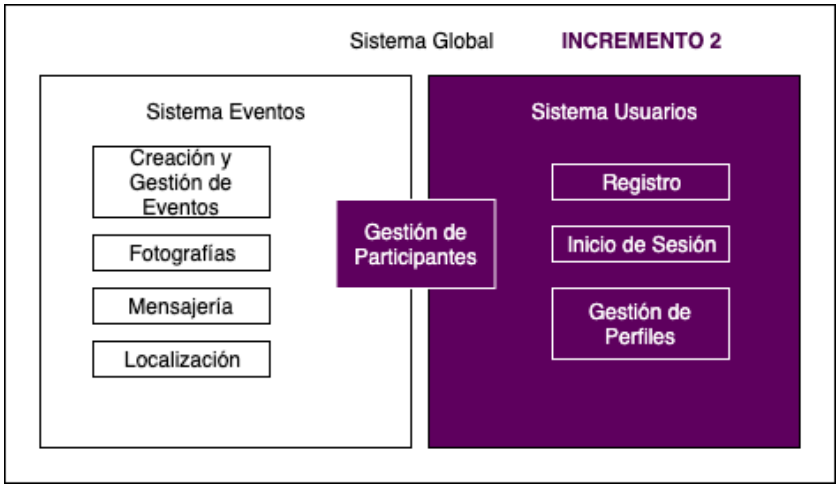


Figura 4.5: Especificación visual de los subsistemas a implementar en el segundo incremento.

Tal y como se ha hecho en la sección anterior, sólo se comentarán las características más relevantes del desarrollo de estos subsistemas. Además del desarollo de estos subsistemas, se hicieron algunas adaptaciones de los subsistemas implementados en el anterior incremento para utilizar los usuarios de la aplicación como, por ejemplo, la adición de los usuarios al subsistema de mensajería para utilizar el nombre de usuario y la imagen de perfil del usuario para mostrar el autor de los mensajes y respuestas.

4.4.1. Desarrollo de los Subsistemas de Registro, Inicio de Sesión y Gestión de Perfiles

En esta subsección se va a explicar el desarrollo de los Subsistemas que conforman el Sistema de Usuarios, es decir, los subsistemas de Registro, Inicio de Sesión y Gestión de Perfiles, se ha decidido hacer un subapartado conjunto debido a que están muy relacionados y tienen una implementación con características similares.

Para el desarrollo de ambos subsistemas tiene especial relevancia la utilización del servicio Firebase de Google. En el apartado 4.2 se explicó el porqué se había decidido utilizar este servicio para

gestionar la autenticación de los usuarios en la aplicación.

El servicio Autenticación de Firebase (Firebase Authentication) provee a los desarrolladores de una herramienta que permite realizar el registro y la autenticación de los usuarios de manera sencilla y segura, haciendo que el desarrollador no tenga que preocuparse de aspectos como la encriptación y almacenamiento de contraseñas. Además, provee de diferentes métodos de registro, en la aplicación sólo se ha implementado el registro y autenticación a través de correo electrónico pero Firebase proporciona métodos para poder autenticarse con otros servicios como cuentas Google, cuentas Facebook, etc.

En este caso, se recogen todos los datos del usuario (nombre, apellidos, nombre de usuario, correo electrónico y contraseña) a través de un formulario creado en la vista de registro de la aplicación.

A continuación, podemos ver un fragmento de código que ejemplifica la sencillez con la que se realiza el registro de un usuario en Firebase Authentication una vez que el usuario ha introducido el correo electrónico y la contraseña con la que quiere realizar el registro.

Código 4.6: Fragmento de código en el que se muestra de forma simplificada el proceso de registro de un usuario en la aplicación

```
1  FirebaseAuth mAuth = FirebaseAuth.getInstance();
2  mAuth.createUserWithEmailAndPassword(email, password)
3          .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
4      @Override
5      public void onComplete(@NonNull Task<AuthResult> task) {
6          if (task.isSuccessful()) {
7              FirebaseUser user = mAuth.getCurrentUser();
8              String userId= user.getUid();
9          }
10     }
11 });
```

Como podemos ver en el fragmento de código, el registro de un usuario en Firebase se reduce a hacer una llamada al método createUserWithEmailAndPassword de la instancia de FirebaseAuth de nuestro sistema.

El código dentro de la función onComplete se ha añadido para mostrar dos características de Firebase Authentication, cuando un usuario se registra en el sistema automáticamente Firebase lo autentica, lo que nos permite recuperar el usuario que acaba de crearse con la función getCurrentUser(). Esta función, que utilizaremos también a la hora de realizar el inicio de sesión, devuelve el usuario que está actualmente autenticado en la instancia de Firebase del sistema. En segundo lugar, se quiere remarcar que es el propio Firebase Authentication el que provee de un identificador único a los usuarios registrados, y que se ha utilizado para identificar al usuario en la base de datos. Este identificador se puede obtener haciendo una llamada a getUid() sobre la instancia de FirebaseUser del usuario logueado.

En cuanto al Inicio de Sesión en la aplicación, este se hace a través de un formulario que solicita al usuario el correo electrónico y la contraseña con la que realizó el registro. Cuando se tienen esos datos la autenticación se realiza de manera sencilla con una llamada al método signInWithEmailAndPassword. Si el inicio de sesión es exitoso podemos recuperar el usuario logueado de la misma forma que hicimos en el registro, con la función getCurrentUser(), para posteriormente recuperar la información completa del usuario en la DB .

Código 4.7: Fragmento de código en el que se muestra de forma simplificada el proceso de inicio de sesión de un usuario en la aplicación

```
1  mAuth.signInWithEmailAndPassword(emailSt,passwordSt).addOnCompleteListener(new
       OnCompleteListener<AuthResult>() {
2      @Override
3      public void onComplete(@NonNull Task<AuthResult> task) {
4          if (task.isSuccessful()) {
5              Log.d(TAG, "logInWithEmail:success");
6              FirebaseUser user = mAuth.getCurrentUser();
7          }
8      }
9  });
```

Por último, el subsistema de gestión de perfiles que permite al usuario cambiar ciertos datos como su nombre y apellidos o su imagen de perfil. Además, permite al usuario Cerrar Sesión o eliminar su cuenta de usuario del sistema. Estas dos últimas funcionalidades, también se realizan utilizando ciertas funcionalidades de Firebase Authentication. El cierre de sesión es un proceso muy sencillo que solo requiere la llamada a la función signOut() de la instancia de Firebase Authentication en la aplicación.

Código 4.8: Fragmento de código en el que se muestra el proceso de cierre de sesión de un usuario en la aplicación

```
1  FirebaseAuth.getInstance().signOut();
```

Por su parte la eliminación de la cuenta, además de eliminar al usuario de Firebase Authentication requiere la eliminación del usuario de la base de datos, tanto el propio objeto del usuario como todas las referencias del mismo en eventos, mensajes y fotografías.

Código 4.9: Fragmento de código en el que se muestra de manera simplificada el proceso de eliminación de un usuario en la aplicación

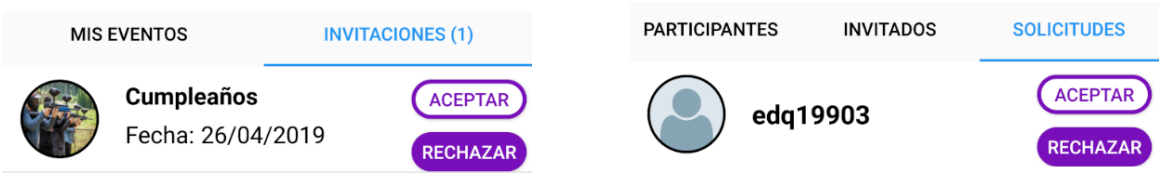
```
1  FirebaseAuth.getInstance().getCurrentUser().delete();
```

4.4.2. Desarrollo del Sistema de Gestión de Participantes

En esta subsección se va a explicar el desarrollo del subsistema de Gestión de Participantes, uno de los subsistemas con mayor importancia debido a que es el nexo de unión entre el sistema de Eventos y el sistema de Usuarios.

Este subsistema se caracteriza principalmente por dos componentes que son los encargados de permitir el acceso de los usuarios a los eventos que son las Invitaciones y las Solicitudes de Acceso. Ambos componentes son muy similares en cuanto a su implementación, su principal diferencia radica en que las invitaciones son enviadas por el organizador a los usuarios para que puedan acceder al evento que organizan. Mientras que las solicitudes las envían los usuarios al evento al que quieren acceder.

La funcionalidad de invitaciones y solicitudes se basa sobre todo en información almacenada en la DB y la presentación en interfaz de esa información. Así una invitación o solicitud consiste principalmente en una entrada en los campos invitations y requests, respectivamente, en los objetos de la DB del evento y el usuario implicados, podemos ver como se presenta esta información en el apartado 4.2.1.



(a) Invitación

(b) Solicitud

Figura 4.6: En la Figura 4.6(a) podemos ver la presentación gráfica de una Invitación. En la Figura 4.6(b) podemos ver la representación gráfica de una solicitud.

Después esa información se presenta de manera gráfica (figura 4.6), en usuarios o en la gestión de participantes del evento. La aceptación de la invitación o solicitud conlleva la eliminación de la misma, así como la actualización de la DB añadiendo al usuario recién admitido al evento y la referencia del evento a los eventos del usuario. Su denegación consiste únicamente en la eliminación de la invitación o solicitud.

5

Pruebas y resultados

Una vez vistos el diseño y el desarrollo llevado a cabo, es el momento de analizar el producto software resultante y las pruebas realizadas sobre el mismo.

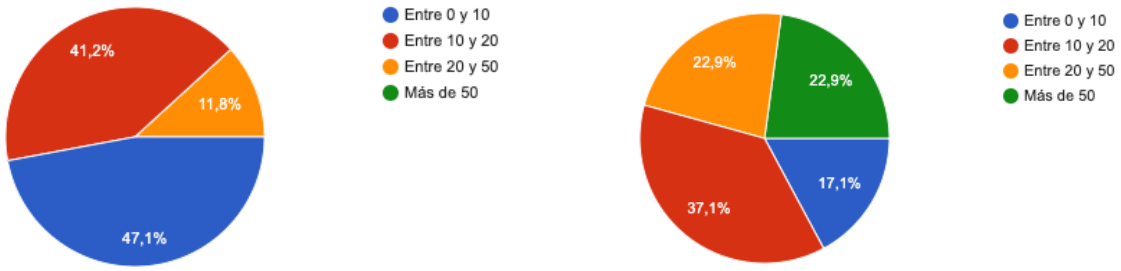
En primer lugar, cabe destacar que una de las pruebas más importantes a las que se sometió el software fue la prueba de campo de la versión beta de la aplicación. En esta prueba de campo, aunque no se tuviera el sistema completo, se pudo comprobar el correcto funcionamiento de la aplicación en un evento real, así como recoger la experiencia de usuarios reales e implementar mejoras y corregir errores.

Cabe destacar que para la versión beta se desarrolló la aplicación tanto para el sistema operativo Android como para el sistema operativo iOS, el desarrollo del segundo incremento de la versión iOS está actualmente en proceso, aunque se ha decidido dejar fuera de este proyecto debido a la falta de tiempo para desarrollar las dos versiones completamente.

El evento en el que se realizó la prueba de campo fue una boda con alrededor de 150 invitados. El APK de Android, se subió a un servicio en la nube que permite a usuarios de testeo descargar e instalar aplicaciones en fase beta a partir de un enlace y una contraseña. En total la versión Android tuvo 105 descargas e instalaciones. Por su parte, la versión iOS no se pudo subir a un servicio en la nube para su distribución debido a las restricciones de Apple, por lo que se optó por instalar la aplicación directamente desde un ordenador a los dispositivos que lo solicitaron, un total de 30 dispositivos.

Para recoger la experiencia de usuario y comprobar si los usuarios seguían usando la aplicación se realizó una breve encuesta a los invitados al evento dos meses después del mismo, la cual fue cumplimentada por 40 usuarios, por lo que lo tomaremos como una muestra representativa. A continuación, veremos algunos de los resultados obtenidos y algunas de las medidas que se tomaron en relación a cada resultado.

En la encuesta se añadieron algunas preguntas para medir el rango de utilización de algunas características y funcionalidades de la aplicación. Una de las características principales de la aplicación es la compartición de fotos entre los participantes. En total, durante el evento y los días posteriores al mismo los participantes subieron un total de 329 fotos al evento.



(a) Subida de Fotografías

(b) Descarga de Fotografías

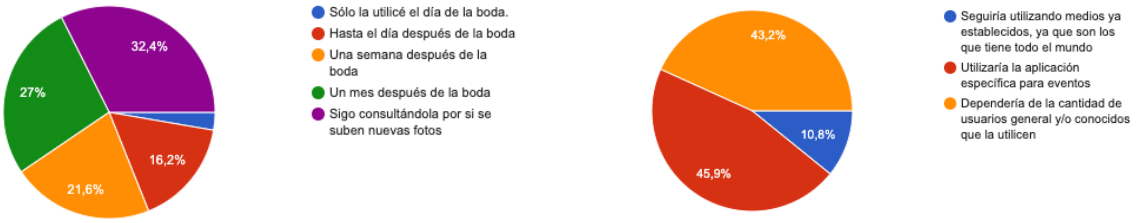
Figura 5.1: En la Figura 5.1(a) podemos ver un gráfico ilustrativo del rango de subida de fotos por participante. En la Figura 5.1(b) podemos ver un gráfico ilustrativo del rango de descarga de fotografías por invitado.

Como podemos ver en los gráficos aproximadamente la mitad de los participantes, no realizarón subidas significativas de fotografías a la aplicación. De los que sí subieron más de 10 fotografías vemos como la mayoría se encuentra en un rango entre 10 y 20 fotografías. Esto ha servido para establecer un límite adecuado de subida simultánea de imágenes que se ha establecido en 20.

En cuanto a la descarga, vemos como más del 80 % de los participantes han descargado más de 10 fotografías de la aplicación y casi un 25 % ha realizado más de 50 descargas de imágenes de la galería del evento. Esto nos lleva a la conclusión, de que es necesario mejorar la funcionalidad de descarga de imágenes y de descarga múltiple. Por lo que para la versión final de la aplicación se decidió incluir un hilo de descarga que permite que la descarga de imágenes se realicé en segundo plano y permite al usuario seguir navegando por la aplicación mientras se realiza la descarga.

En cuanto al tiempo de uso (figura 5.2(a)) y la preferencia de utilización de una aplicación específica de eventos frente a otros medios ya establecidos como Facebook, Whatsapp, etc. (figura 5.2(b)), nos encontramos que un porcentaje bastante elevado de participantes seguía consultando el evento de manera eventual después de dos meses por sí había nuevas fotografías. Y un 80 % siguió consultandolo después de una semana de la realización del evento. Esto es significativo ya que nuestros usuarios no se centran sólo en la organización previa al evento y en el día del evento, sino que se centra también en el tiempo posterior al mismo donde el consumo y subida de imágenes sigue siendo bastante significativo.

En cuanto a la preferencia de los usuarios de utilizar una aplicación específica frente a las ya consolidadas o convencionales, parece que una gran parte de los usuarios preferiría una aplicación centrada en el evento mientras que otra gran parte opina que dependería de la cantidad de usuarios que tuviera la aplicación centrada en eventos. Lo que si podemos abstraer de este resultado es que los usuarios no parecen estar cerrados a incorporar nuevas redes sociales a sus redes habituales y que, aunque con condiciones, puede haber un mercado abierto a este tipo de aplicación.



(a) Tiempo de utilización

(b) Aplicacion centrada en Eventos vs. Medios convencionales

Figura 5.2: En la Figura 5.2(a) podemos ver un gráfico ilustrativo del rango de tiempo durante el cual los participantes han seguido utilizando la aplicación. En la Figura 5.2(b) podemos ver un gráfico ilustrativo la predisposición de los usuarios de utilizar una aplicación centrada en eventos frente a los medios ya establecidos.

Por último, otro de los aspectos sobre los que se pidió la opinión a los usuarios fue sobre el diseño y la funcionalidad que ofrecía la beta. En general, los resultados fueron buenos tanto en diseño como en funcionalidad con una media de puntuación (entre 0 y 10) de 8,16 y 8 respectivamente. Además, se solicitó a los usuarios que dijeran qué mejoras añadirían a la aplicación, lo cual sirvió para implementar mejoras como cambios en la interfaz, la posibilidad de navegar por la galería dentro del modo de visualización detallada, comentarios en las fotografías, etc.

Como ya se ha comentado a lo largo de esta memoria, uno de los objetivos principales en el desarrollo de la aplicación es que ésta fuera eficiente tanto en consumo de memoria RAM, utilización de la CPU y en el consumo de datos. Android Studio provee de una herramienta que permite ver el consumo de la aplicación en estos aspectos. A continuación, se van a mostrar los resultados obtenidos con un evento de prueba creado con 100 imágenes.

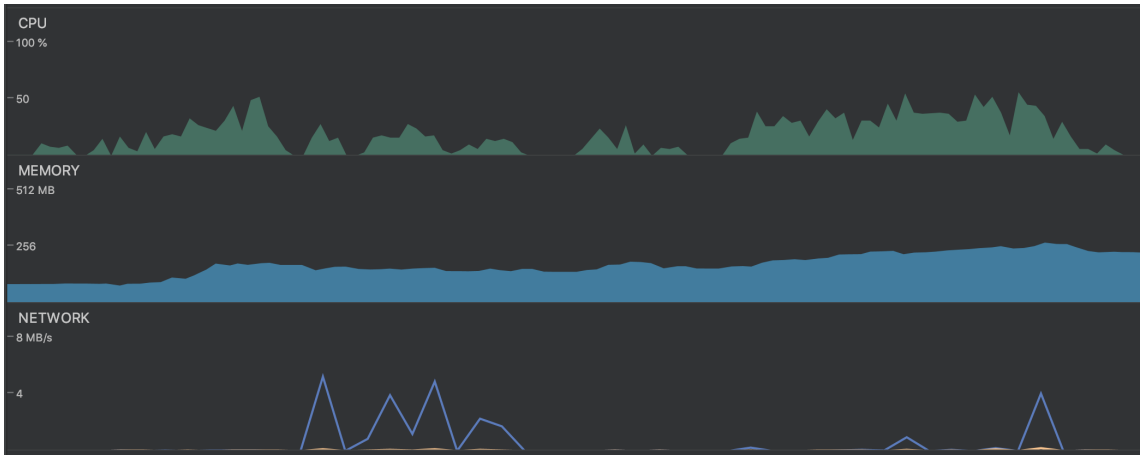


Figura 5.3: Grafíca de consumo de recursos en un evento de prueba

Como podemos ver en la figura 5.3, existen dos momentos en los que se producen picos tanto en el uso de CPU como en el uso de red. El primero se corresponde con el login del usuario en la

aplicación, donde se deben descargar de la base de datos tanto el propio usuario como los eventos a los que pertenece. El segundo pico se produce cuando el usuario accede a la galería del evento y se descargan las previsualizaciones de las imágenes o thumbnail's. Sin embargo, podemos ver como esos picos se mantienen dentro de un rango adecuado. En ningún momento se supera el 50 % de uso de CPU del dispositivo, la memoria RAM consumida es muy estable con un pico máximo de 300 MB, pero situándose en torno a los 256 MB de media. Por último, en red vemos como existen picos pero en ningún momento se alcanzan los 8 MB/s. El consumo de datos dependerá al final del número de fotografías que el usuario visualice detalladamente y descargue en su dispositivo, pero se ha logrado controlar el consumo medio de datos y los recursos consumidos por la aplicación son más que aceptables.

Por último, vamos a dar algunas estadísticas en Líneas de Código (LDC) para que el lector pueda hacerse una idea aproximada de la magnitud del sistema implementado:

- **Líneas de Código Java:** 12330 LDC
- **Líneas de Código XML:** 7205 LDC
- **Total LDC implementado:** 19535 LDC

La implementación del sistema total ha llevado aproximadamente 6 meses de trabajo, incluyendo el diseño, la implementación y pruebas del sistema.

6

Conclusiones y trabajo futuro

6.1. Conclusiones

En esta sección vamos a hacer un breve análisis de los resultados obtenidos a modo de conclusión. Como se ha podido observar a lo largo de la memoria, el sistema que se ha desarrollado es muy amplio y contiene una gran cantidad de funcionalidad.

El objetivo general del proyecto era crear una Red Social de Eventos Android que permitiera a los usuarios crear y gestionar eventos, así como compartir mensajes y fotografías entre los participantes de los eventos. Además, se buscaba que la aplicación fuera lo suficientemente atractiva y con una funcionalidad potente con la finalidad de que pudiera tener una oportunidad en el mercado actual de redes sociales. A raíz de los resultados obtenidos, considero que este objetivo final ha sido logrado con éxito. El producto software resultante, además de ser viable y con gran potencial competitivo en el mercado, ha sido evaluado positivamente por los usuarios que han testeado la aplicación en su versión beta. Además, las pruebas de rendimiento realizadas son altamente positivas.

Además, gracias al diseño realizado para el desarrollo del proyecto el software es fácilmente ampliable con nuevas funcionalidades y módulos que puedan hacer del producto una red social más atractiva para los usuarios.

6.2. Trabajo futuro

Como ya se ha comentado en los resultados, actualmente se está trabajando en el desarrollo del segundo incremento de la versión iOS de la aplicación. El objetivo es poder lanzar al mercado (Play Store y App Store) ambas versiones a finales del verano de 2019. Sin embargo, el camino a recorrer para lograr este objetivo es largo ya que todavía quedan muchos flecos por cubrir en el desarrollo de la red social. Uno de los objetivos antes del lanzamiento, a parte del desarrollo de la versión iOS, debe ser analizar la mejor manera de obtener beneficios con el lanzamiento de la app para que sea un producto económicamente viable, bien con la introducción de publicidad en la aplicación como hacen otras redes sociales o mediante la posibilidad de que comercios locales (bares, discotecas, salas de

conciertos, etc.) puedan dar publicidad a los eventos que organicen. En cualquier caso, la aplicación requerirá de ajustes para adaptar estos cambios.

Así mismo, personalmente me gustaría seguir ampliando y mejorando la funcionalidad que ofrece EventSharing, con nuevos módulos como la compartición de ubicación entre usuarios, mensajería instantánea entre los usuarios, etc., por lo que las posibilidades de ampliación son muy variadas.

Bibliografía

[1] M. Ros-Martín, “Evolución de los servicios de redes sociales en internet,” *El profesional de la información*, vol. 18, no. 5, pp. 552–557, 2009.

[2] IAB Spain, “Estudio Anual Redes Sociales 2018,” IAB Spain, Tech. Rep., 2018. [Online]. Available: https://iabspain.es/wp-content/uploads/estudio-redes-sociales-2018{_}vreducida.pdf

[3] Faceebook, “Información de la Empresa,” 2019. [Online]. Available: https://es.newsroom.fb.com/company-info/

[4] Europapress, “Google Maps introduce la posibilidad de crear eventos públicos,” mar 2019. [Online]. Available: https://www.europapress.es/portaltic/internet/noticia-google-maps-introduce-posibilidad-crear-eventos-publicos-20190325175918.html

[5] D. Longstreet, “Fundamentals of Function Point Analysis,” 2005. [Online]. Available: http://www.softwaremetrics.com/files/FundamentalsofFunctionPointAnalysis.pdf

[6] Google Developers, “Android Studio,” 2019. [Online]. Available: https://developer.android.com/studio

[7] Google Developers , “Firebase,” 2019. [Online]. Available: https://firebase.google.com/products/?nav=true

[8] Google, “PlaceAutocompleteAdapter.java,” 2015. [Online]. Available: https://github.com/googlesamples/android-play-places/blob/master/PlaceCompleteAdapter/Application/src/main/java/com/example/google/playservices/placecomplete/PlaceAutocompleteAdapter.java

Definiciones

actividad Una actividad es un componente de la aplicación que contiene una vista con la que interactúan los usuarios. A diferencia de los fragmentos, las actividades no pueden estar contenidas en otras actividades sino que se cambia de una actividad a otra.

adaptador En Android, un adaptador es una clase especial de componentes que sirve para especificador el comportamiento o la lógica de un componente determinado dentro de la aplicación. Por ejemplo, las listas utilizan adaptadores para especificar qué datos deben pintar o el comportamiento al pulsar sobre un elemento de la lista.

API *Application Programming Interface*, es un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. El número de una API (p.ej. API 21) especifica la versión de la misma.

APK Un archivo con extensión .apk contiene una aplicación compilada para el sistema operativo Android. El APK sirve como instalador de la aplicación en el dispositivo Android y contiene todos los archivos para instalar la app y ejecutarla.

fragmento Un fragmento es un componente de la interfaz que se "incrusta" en una Actividad, permite presentar múltiples vistas en una misma interfaz de usuario de manera que permite agregar, quitar o sustituir partes de la interfaz dentro de la misma vista.

organizador Usuario registrado que ha creado un evento, y del cuál es administrador.

participante Usuarios perteneciente a un determinado evento, incluido el organizador del mismo. Para ser participante del evento el usuario debe haber aceptado una invitación de acceso al evento o el organizador debe haber aceptado una solicitud de acceso del usuario al evento.

RecyclerView RecyclerView es un tipo especial de vista de Android que permite presentar listas o cuadriculas deslizables verticalmente (scrolleables).

swipe Gesto que se realiza deslizando un dedo sobre la pantalla del dispositivo hacia la derecha o la izquierda.

thumbnail Un thumbnail o miniatura es una versión reducida de una imagen. Sus usos pueden ser distintos desde la utilización de la miniatura para reconocer una carpeta hasta la optimización de descargas de imágenes.

Acrónimos

DB Base de Datos.

LDC Líneas de Código.

RRSS Redes Sociales.

TDI Grado Total de Influencia.

VAF Valor del Factor de Ajuste.

Apéndices

A

Cálculo de Puntos de Función

En este apartado se va a desarrollar detalladamente la estimación por puntos de función realizada para este proyecto. Para llevar a cabo esta tarea se han seguido los estándares establecidos por IFPUG, resumidos por Longstreet [5].

En primer lugar, se va a catalogar cada uno de los requisitos funcionales del sistema en uno de los siguientes Tipos de Funciones de Transacción:

- **Entrada Externa (EI):** Proceso elemental en el que los datos cruzan la frontera desde el exterior al interior del sistema donde son registrados, para simplificar se trata de entrada de datos al sistema.
- **Salida Externa (EO):** Proceso elemental en el que se exportan datos fuera del sistema que han sido calculados o derivados por algún tipo de proceso del sistema.
- **Consulta Externa (EQ):** Proceso elemental que tiene tanto componentes de entrada como de salida, sin embargo en este caso los datos de entrada no son registrados por el sistema y los datos de salida no han sido procesados de ninguna manera por el sistema.

A parte de las Funciones de Transacción que definen la entrada y salida de datos de nuestro sistema, debemos establecer los archivos lógicos de datos donde se almacenarán y desde donde se recuperarán los datos. Encontramos dos tipos diferentes de archivos:

- **Archivo Lógico Interno (ILF):** Es un conjunto identificable de datos relacionados que residen y son mantenidos enteramente dentro de los límites de nuestro sistema, a través de EI's.
- **Archivo de Interfaz Externo (EIF):** Es un conjunto identificable de datos relacionados que se referencian desde nuestro sistema pero que residen en su totalidad fuera de los límites de nuestro sistema y son mantenidos por otra aplicación.

Además de catalogar los requisitos funcionales en uno de los tres tipos de proceso elemental, debemos establecer la complejidad de cada componente o requisito en función del número de archivos con los que interactúan y el número de tipos elementales de datos que representan. Esto es en resumen, establecer con cuantos almacenes de datos interactúan y la cantidad de datos que actualizan o recuperan cada uno de los requisitos.

Una vez que estén catalogados y determinada su complejidad ya podremos establecer los puntos de función desajustados que se otorgan a cada requisito, siguiendo la siguiente tabla de asignación (figura A.1).

Tipo de Componente	Complejidad Baja (B)	Complejidad Media (M)	Complejidad Alta (A)
Entrada Externa (EI)	3	4	6
Salida Externa (EO)	4	5	6
Consulta Externa (EQ)	3	4	6
Archivo Lógico Interno (ILF)	7	10	15
Archivo Lógico Externo (EIF)	5	7	10

Figura A.1: Tabla resumen en la cual se especifica los puntos de función que deben asignarse a cada componente en función de su tipo y su complejidad

Con esta información, ya podemos calcular los puntos de función de cada uno de nuestros requisitos, tal y como se muestra en la tabla de la figura A.3.

Una vez que se han obtenido los puntos de función de cada requisito funcional debemos calcular también los puntos de función asociados a cada uno de los archivos de datos que se van a utilizar en nuestro sistema (figura A.2).

COMPONENTE	TIPO	COMPLEJIDAD	PUNTOS DE FUNCIÓN
Usuarios	ILF	A	10
Eventos	ILF	A	15
Fotografías	ILF	B	7
Mensajes	ILF	B	7
Google Places	EIF	M	7
Google Maps	EIF	B	5

Figura A.2: Tabla en la que se especifica el tipo, la complejidad y los puntos de función resultantes de cada uno de los archivos de datos utilizados por el sistema.

SUBSISTEMA	Ref. Requisito	Tipo de Transacción	Complejidad	Puntos de Función
REGISTRO	RF-1	EI	M	4
	RF-2	EO	B	4
	RF-3	EQ	B	3
	RF-4	EI	M	4
	RF-5	EQ	B	3
	RF-6	EO	B	4
INICIO DE SESIÓN	RF-7	EQ	B	3
	RF-8	EI	M	4
	RF-9	EO	B	4
	RF-10	EI	B	3
GESTIÓN DE PERFILES	RF-11	EI	B	3
	RF-12	EQ	M	4
	RF-13	EQ	M	5
	RF-14	EO	B	4
	RF-15	EQ	M	4
	RF-16	EI	B	3
	RF-17	EI	A	6
CREACIÓN Y GESTIÓN DE EVENTOS	RF-18	EI	A	6
	RF-19	EO	M	5
	RF-20	EO	B	4
	RF-21	EI	B	3
	RF-22	EQ	M	4
GESTIÓN DE PARTICIPANTES	RF-23	EQ	M	4
	RF-24	EQ	M	4
	RF-25	EQ	B	3
	RF-26	EO	B	4
	RF-27	EQ	B	3
MENSAJERÍA	RF-28	EI	B	3
	RF-29	EI	B	3
	RF-30	EQ	B	3
	RF-31	EQ	B	3
FOTOGRAFÍAS	RF-32	EO	M	5
	RF-33	EQ	B	3
	RF-34	EO	B	4
	RF-35	EQ	B	3
	RF-36	EQ	M	4
	RF-37	EI	B	3
	RF-38	EQ	M	4
LOCALIZACIÓN	RF-39	EQ	B	3
	RF-40	EO	B	4

Figura A.3: Tabla en la que se especifica el tipo, la complejidad y los puntos de función resultantes de cada uno de los requisitos funcionales del sistema.

Con estos datos, ya tenemos la totalidad de puntos de función desajustados que componen nuestro sistema. Sin embargo, para que los cálculos sean más precisos debemos ajustar los puntos de función obtenidos a partir de las características propias de nuestro sistema. Para ello debemos obtener el valor del factor de ajuste (VAF, de sus siglas en inglés *Value Adjustment Factor*). Para obtener el VAF debemos establecer la configuración de nuestro sistema estableciendo una serie de valores a diferentes caraceterísticas del sistema, de esta manera obtendremos el Grado Total de Influencia (TDI, de sus siglas en inglés *Total Degree of Influence*). En la figura A.4 podemos observar los valores establecidos para la configuración del sistema.

$$VAF = 0{,}65 + (\frac{TDI}{100}) \tag{A.1}$$

Siendo TDI,

$$TDI = \sum_{i=1}^{14} C_i \tag{A.2}$$

Características Generales del Sistema	Valoración
Comunicación de Datos	3
Procesamiento de Datos Distribuidos	4
Prestaciones	3
Gran Uso de la Configuración	1
Velocidad de Transacciones	4
Entrada de datos online	5
Eficiencia del Usuario	4
Actualización de Datos Online	4
Procesamiento Complejo	1
Reusabilidad	2
Facilidad de Instalación	1
Facilidad de Operación	1
Múltiples Localizaciones	3
Facilidad de Cambio	3
Grado de Influencia (TDI)	**39**

Figura A.4: Tabla en la que se especifica el grado de influencia en el sistema de distintas características generales del sistema

Usando el valor de TDI obtenido en la figura A.4 y la ecuación A.1, obtenemos el siguiente valor:

$$VAF = 0{,}65 + 0{,}39 = 1{,}04 \tag{A.3}$$

Con este valor podemos aplicar la ecuación A.4 para obtener los puntos de función ajustados.

$$PF_{Ajustados} = PF_{Desajustados} \times VAF \tag{A.4}$$

Estos puntos de función ajustados nos otorgan un recurso preciso para estimar la complejidad y tamaño del sistema a desarrollar, además nos permitirán realizar una planificación temporal del proyecto mucho más precisa.

B

Base de Datos de la Aplicación

En este apéndice se explicarán los diferentes objetos JSON creados para el almacenamiento de la información de la aplicación en la base de datos de Firebase. El objetivo de este apéndice es dar una idea más detallada de la implementación de la base de datos y su funcionamiento.

Evento

En primer lugar, vamos a explicar el objeto JSON en el que se almacenan los eventos de la aplicación.

Código B.1: Objeto JSON asociado a los eventos en la DB

```
"events": {
  "-LbF-mvIipSg7JeY0ULs" : {
    "adminIds" : [ "jnJsUmo6WqRUEVZPfbQ19c98PID2" ],
    "date" : "26/04/2019",
    "description" : "Para celebrar mis 29 añazos he ... ",
    "eventDBID" : "-LbF-mvIipSg7JeY0ULs",
    "eventID" : "edq19903-Cumpleaños",
    "eventTitle" : "Cumpleaños",
    "invitationIds" : {
      "Gey9k1ScjyUyiNngRqhNo5UpX5i1" : "-LbF-mvYnpN6S2vs9dzt"
    },
    "mainImage" : "https://firebasestorage.googleapis.com/v0/b/tfg2018-2019.appspot.com/...",
    "participantsIds" : {
      "tU53nZ1AzANL4i6GiNdT5FS6IvT2" : "tU53nZ1AzANL4i6GiNdT5FS6IvT2"
    },
    "position" : {
      "latitude" : 40.4167396,
      "longitude" : -3.7031921
    },
    "requests" : {
      "-LR6on_kIm0_4qp4ouE0" : "tU53nZ1AzANL4i6GiNdT5FS6IvT2"
    },
```

Como podemos observar en el código, los eventos se guardan en el objeto “eventos”. Cada evento se guarda dentro de este objeto con el identificador único que le asigna Firebase en la subida a la

DB , que será el identificador utilizado para recuperar el evento de la DB . Dentro de esta estructura, encontramos todas las propiedades del evento que deben ser almacenadas para su posterior recuperación: el identificador del organizador, fecha, descripción del evento, el identificador del evento en la DB , el identificador del evento en la aplicación, el título del evento, los identificadores de las invitaciones pendientes del evento, la URL de la imagen de cabecera del evento, los identificadores de los participantes, los datos de localización del evento (latitud y longitud), los identificadores de las solicitudes pendientes que en el evento se almacenan con el identificador de la solicitud como etiqueta y el identificador del usuario que la ha realizado como valor y la hora de realización.

Usuarios

Este objeto JSON se ha diseñado e implementado para almacenar y recuperar la información relativa a los usuarios.

Código B.2: Objeto JSON asociado a los usuarios en la DB

```
"users":{
    "Tay4k1SfjyUyiNsgRqhNo2UpX3i2" : {
        "email" : "e.dequinto103@gmail.com",
        "events" : {
          "-LPLrDdF-sM7Pg0vbsPD" : "-LsLgDdF-sM74g0vbsPD",
          "-LQdbaUj7LgyAAa7v6ai" : "-LQsBaUj7Lgyq3Aa7v6i"
        },
        "invitations" : {
          "-LRudWB5qt9UhIPwCfb3" : {
            "eventId" : "-LQaMjYrd56jNR3uP_oN",
            "invitationId" : "-LRudWB5qt9UhIPwCfb3"
          }
        },
        "lastName" : "de Quinto",
        "name" : "Edu",
        "nickname" : "edq2",
        "photo" : "https://firebasestorage...",
        "requests" : {
            "-La79IrOhUz1qc2nLReZ" : {
              "eventID" : "-LZMZ4fRIWe6nxjGKq43",
              "requestID" : "-La79IrOhUz1qc2nLReZ",
              "userID" : "Tay4k1SfjyUyiNsgRqhNo2UpX3i2"
            }
          },
        "userId" : "Tay4k1SfjyUyiNsgRqhNo2UpX3i2"
      }
}
```

Como podemos observar en el código, cada usuario se almacena en un objeto que tiene como etiqueta principal un identificador que es único y es asignado por Firebase cuando un usuario se registra en la aplicación. En cada usuario se almacena la siguiente información: correo electrónico con

el que realizó el registro, eventos a los que pertenece identificados por su identificador único en la DB , invitaciones pendientes que tienen un identificador único asignado por la DB y están compuestas por el identificador del evento y el de la propia invitación; apellidos y nombre del usuario, nombre de usuario, URL de la foto de perfil, solicitudes de acceso pendientes identificadas por un identificador único de la solicitud asignado por Firebase y están compuestas por el identificador del evento, el identificador del usuario y el identificador de la propia solicitud y el propio identificador del usuario en la DB .

Mensajes y Respuestas

Otro de los objetos que encontramos en la DB son los mensajes. A continuación podemos ver la estructura de los mensajes.

Código B.3: Objeto JSON asociado a los mensajes en la DB

```
"messages" : {
  "-LQdbaUj7LgyAAa7v6ai" : {
    "-LTaoEYb-FzhhNGlTM20" : {
      "admin" : true,
      "answers" : {
        "-LTaokOuSaqSH3KuHg6t" : {
          "answer" : "Holaa espero que todos vengáis",
          "userID" : "326mwjhtKif7K3RzQFZ217a5G3nU2"
        }
      },
      "message" : " Holaaaaa otra veez",
      "user" : "Tay4k1SfjyUyiNsgRqhNo2UpX3i2"
    }
  }
}
```

En primer lugar vemos que en este caso hay dos etiquetas con identificadores al principio del objeto, la primera es el identificador del evento al que pertenece el mensaje y la segunda es la etiqueta principal de cada mensaje que es un identificador único del mensaje asignado por Firebase, el contenido del objeto es el siguiente: campo booleano que determina si es un mensaje del organizador o no, respuestas del mensaje cada una de las cuales tiene como etiqueta principal un identificador único de la respuesta y el contenido del objeto es el contenido de la respuesta y el identificador del usuario que la ha creado; el contenido del mensaje y el identificador del usuario que ha creado el mensaje.

Fotografías

Por último, vamos a ver el objeto JSON en el que se almacenan las fotografías del sistema.

La fotografía se almacena de manera similar a los mensajes, como podemos observar tiene dos etiquetas con identificadores en primer lugar, las cuáles son el identificador del evento al que pertenece la fotografía, y el propio identificador único de la fotografía en la base de datos como etiqueta principal

Código B.4: Objeto JSON asociado a las fotografías en la DB

```
"photos":{
    "-LQdbaUj7LgyAAa7v6ai" : {
        "-LQe-udMOrgHcSyzPexC" : {
          "date_created" : "2018-11-06T18:13:24Z",
          "image_id" : "-LQe-udMOrgHcSyzPexC",
          "image_path" : "https://firebasestorage.googleapis.com/v0/b/tfg2018...",
          "preview_path" :
                "https://firebasestorage.googleapis.com/v0/b/tfg2018-2019.appspot.com/..."
        }
    }
}
```

de la misma. Dentro del objeto podemos encontrar la siguiente información: fecha de subida de la imagen, identificador de la imagen, URL de la imagen en tamaño completo y URL de la imagen en tamaño reducido o thumbnail.

Cabe destacar que tanto fotografías y mensajes forman parte de los eventos, sin embargo se ha decidido crear objetos independientes para su almacenamiento, en lugar de almacenarlos dentro del propio objeto del evento al que pertenecen, ya que los eventos son objetos que se descargan de la DB en múltiples vistas y se ha querido reducir el tamaño de estos objetos. Así si una vista requiere de información del evento como el título, o el organizador o sus participantes, pero no requiere los mensajes ni las fotografías, no tendrá que descargar esta información reduciéndose el tamaño de las descargas de información de la DB .

C

Manual de Usuario

A continuación, se va a detallar toda la funcionalidad disponible en la aplicación, así como la interfaz gráfica desarrollada. El objetivo de este apéndice es proveer al lector de un manual de usuario de la aplicación para que pueda utilizar todas las características que aporta EventSharing.

La primera vista con la que se encontrará el usuario al acceder a la aplicación es la vista de Inicio de Sesión(figura C.1), esta vista permite a los usuarios ya registrados acceder a la aplicación introduciendo su correo electrónico y su contraseña. Además tiene dos botones, el primero de ellos, etiquetado como “Registro”, cambia a la vista de Registro (figura C.2) que permite al usuario crear una cuenta nueva en la aplicación. El segundo etiquetado como “¿Olvidaste la contraseña?” cambia a la vista de recuperación de contraseña (figura C.4).

En caso de que el usuario introduzca un correo electrónico y contraseña válidos se accede a la aplicación y se muestra la vista principal del usuario.

Figura C.1: Vista de Inicio de Sesión

La vista de Registro (figura C.1), consiste básicamente en un formulario que el usuario debe rellenar

para poder crear una cuenta nueva en la aplicación.

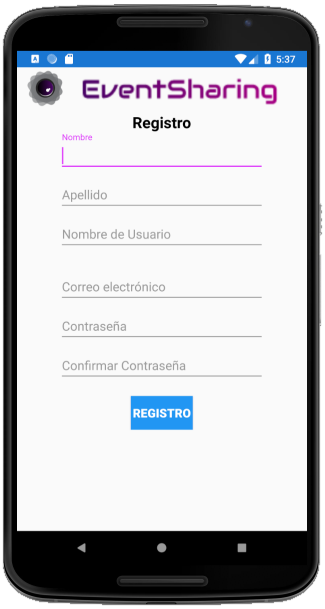


Figura C.2: Vista de Registro que permite a un usuario crear una cuenta en la aplicación

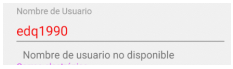
Se han introducido una serie de restricciones, que en caso de no cumplirse mostrarán diferentes mensajes de error al usuario:

- **Nombre de usuario único:** El nombre de usuario elegido por el usuario debe estar disponible, es decir, no puede estar actualmente en uso por otro usuario.
- **Formato de nombre de usuario:** El nombre de usuario debe tener entre 3 y 15 caracteres, sin mayúsculas, ni símbolos ni espacios.
- **Correo electrónico válido:** El correo electrónico introducido debe ser válido.
- **Cuenta única:** El correo electrónico introducido no debe estar actualmente en uso por otra cuenta en la aplicación.
- **Contraseña fuerte:** La contraseña introducida debe ser una contraseña fuerte, es decir, debe tener entre 8 y 16 caracteres, y contener un número y una mayúscula.
- **Confirmación de contraseña:** La contraseña debe ser introducida dos veces y ambas deben coincidir.

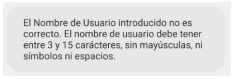
En la figura C.3, podemos ver los errores que se muestran en la aplicación cuando no se cumple cada una de estas restricciones.

En cuanto a la vista de Recuperación de Contraseña (figura C.4), permite a un usuario restablecer su contraseña cuando se ha olvidado y no puede acceder a su cuenta para cambiarla desde dentro. En este caso, el usuario debe introducir el correo electrónico con el que se registró y se le envía un correo electrónico de restablecimiento de contraseña, permitiéndole acceder de nuevo a la aplicación con la nueva contraseña.

Una vez que el usuario se ha logueado correctamente en la aplicación, lo primero que vera será la vista principal del usuario (figura C.5) de la cual ya tuvimos un primer vistazo en el apartado 3.3. Esta



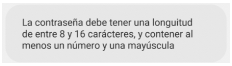
(a) Nombre de usuario no disponible



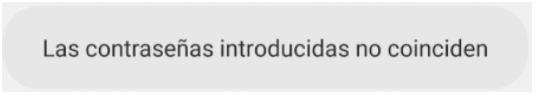
(b) Nombre de usuario inválido

(c) Correo electrónico inválido

(d) Correo electrónico en uso



(e) Contraseña débil



(f) Confirmación de contraseña

Figura C.3: En la figura C.3(a) podemos ver como se avisa al usuario a medida que va escribiendo su nombre de usuario de la disponibilidad del mismo. En la figura C.3(b) se puede ver el mensaje de error cuando se introduce un usuario inválido. En la figura C.3(c) se puede ver el mensaje de error al introducir un correo electrónico inválido. En la figura C.3(d) se puede ver el mensaje de error al introducir un correo actualmente en uso por otra cuenta. En la figura C.3(e) se puede ver el mensaje de error al introducir una contraseña débil. En la figura C.3(f) se puede ver el mensaje de error si no coinciden las contraseñas introducidas en el campo “Contraseña” y “Confirmar Contraseña”.

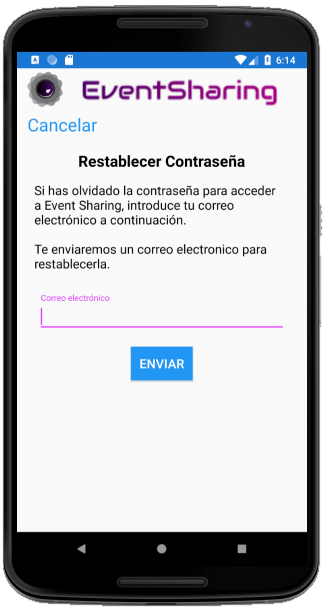


Figura C.4: Vista de Restablecimiento de Contraseña que permite a un usuario cambiar la contraseña de su cuenta en caso de olvido

vista tiene 4 pestañas diferentes que podemos ver en la barra de navegación inferior. La primera de ellas es la vista de eventos del usuario (visible en la figura C.5). Esta vista a su vez está dividida en dos pestañas que podemos ver en la barra de navegación superior etiquetadas como “MIS EVENTOS” y “INVITACIONES”. En la primera de ellas, se puede observar una lista en la que aparecen todos los eventos en los que participa y/o organiza actualmente el usuario. Al pulsar sobre cualquiera de estos eventos se accede al contenido del evento cargando la vista principal del evento (figura C.11). Además, como podemos observar en el segundo evento de la lista, si el usuario ha solicitado acceso a un evento y aún no ha sido respondido aparecerá el evento con el mensaje “Solicitud de acceso pendiente”, y en caso de pulsar sobre ese ítem aparecerá un mensaje indicando al usuario que todavía no tiene acceso al evento.

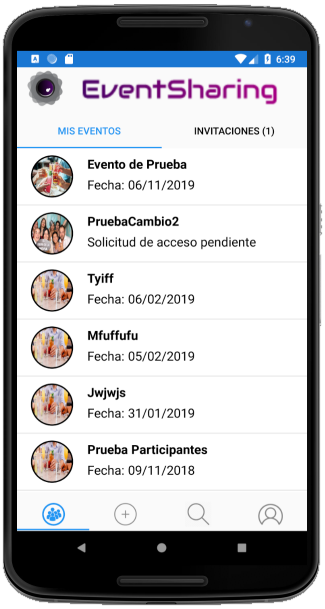


Figura C.5: Vista de principal del usuario.En la vista principal se está mostrando la lista de eventos a los que pertenece el usuario

La segunda de las vistas de eventos del usuario, se corresponde con las invitaciones pendientes de respuesta del usuario (figura C.6). Cada ítem de este listado tiene dos botones uno para aceptar la invitación y otro para rechazarla. En caso de pulsar el primero, el evento se añade a los eventos del usuario y podrá acceder al contenido. En caso contrario, desaparece la invitación y el usuario no podrá acceder al contenido del evento.

En la segunda pestaña de la ventana principal, el usuario puede crear un nuevo evento en la aplicación. Este proceso está formado por 3 vistas diferentes que recogen toda la información necesaria para crear un evento. La primera solicita al usuario el título, la descripción, la fecha, la hora y la imagen de cabecera del evento (figura C.7(a) y figura C.7(b)). En la selección de imagen de cabecera el usuario puede seleccionar una de las imágenes de muestra que ofrece la aplicación o seleccionar una de su propia galería pulsando “Añadir Imagen”.

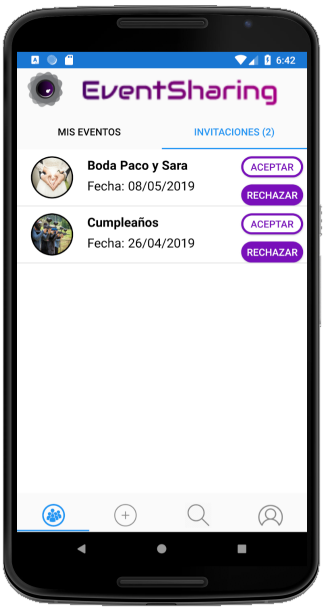
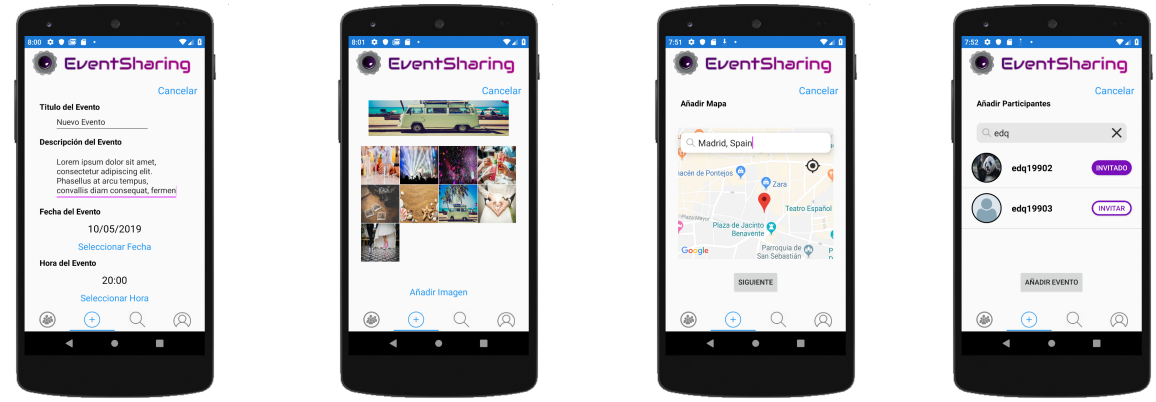


Figura C.6: Vista de invitaciones del usuario, en la que aparece un listado de invitaciones recibidas por el usuario para participar en eventos.

En la segunda parte del proceso de creación (figura C.7(c)), el usuario debe introducir la ubicación en la que se celebrará el evento. La ubicación puede introducirse buscando una dirección o punto de interés en la barra de búsqueda o navegando en el mapa y pulsando donde quiera establecer la ubicación. Por último, el tercer paso de creación (figura C.7(d)) permite al usuario invitar a otros usuarios al evento. Si el usuario comienza a escribir el nombre de usuario de otros usuarios en la barra de búsqueda, empezarán a aparecer coincidencias en la lista situada bajo la barra. Cada ítem tiene un botón que permite invitar al usuario. Si el usuario ya ha sido invitado al evento, este botón cambia de "Invitar" a "Invitado". Si se volviera a pulsar el botón se cancelaría la invitación y el botón volvería a su estado original.

La tercera pestaña de la vista principal se corresponde con la vista de Búsqueda de Eventos (figura C.8), en la cual los usuarios pueden buscar eventos por título, nombre de usuario del organizador, o por el identificador del evento. A medida que el usuario va escribiendo en la barra de búsqueda, los resultados con coincidencias en el alguno de estos campos van apareciendo en el listado inferior. Los eventos a los que ya pertenece el usuario aparecen sin ningún botón, y al pulsar sobre un ítem al que se pertenece se accede al contenido del evento pulsado. Por su parte, los eventos a los que no se pertenece y a los cuales no se ha sido invitado, aparecen junto con un botón que permite al usuario solicitar el acceso a los mismos. Al pulsar el botón, se enviará la solicitud de acceso y se incluirá el evento entre los eventos del usuario (figura C.5), con el mensaje "Solicitud de acceso pendiente".



(a) Paso 1.1. (b) Paso 1.2. (c) Paso 2 (d) Paso 3

Figura C.7: En la figura C.7(a) podemos ver parte del primer paso en la creación del evento donde se solicita el título, la descripción, la fecha y la hora del evento. En la figura C.7(b) podemos ver la segunda parte del primer paso de creación del evento donde se pide al usuario que selecciones una imagen de cabecera para el evento. En la figura C.7(c) podemos ver el segundo paso de creación del evento en el que se solicita al usuario que establezca una localización al evento. En la figura C.7(d) podemos ver el tercer y último paso de creación del evento en el que el usuario puede invitar a otros usuarios al evento.

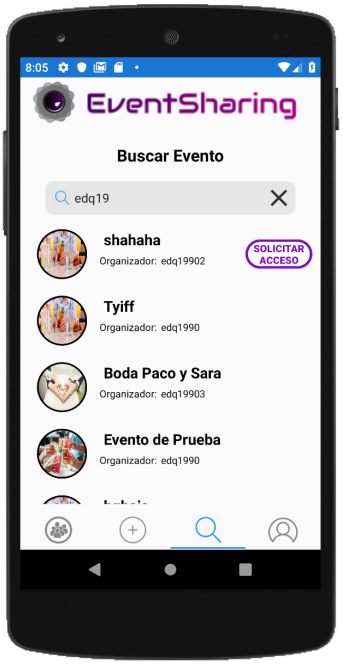
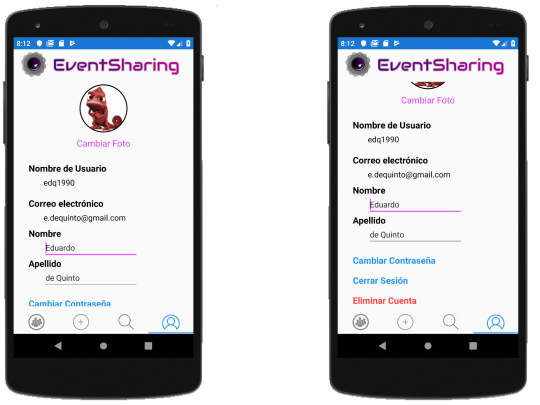


Figura C.8: Vista de Búsqueda de Eventos, formada por una barra de búsqueda y una lista de resultados.

En cuanto a la cuarta y última vista de la ventana principal, se corresponde con la vista de Perfil de Usuario (figura C.9). En esta vista el usuario puede consultar y/o modificar algunos datos de su perfil, como la imagen de perfil, el nombre de usuario, el correo electrónico, el nombre y los apellidos. Cabe mencionar que ni el correo electrónico ni el nombre de usuario son modificables. Debajo de estos campos, el usuario dispone de tres botones, el primero de ellos etiquetado como “Cambiar Contraseña” permite al usuario modificar su contraseña introduciendo la contraseña actual y una contraseña nueva. El segundo, etiquetado como “Cerrar Sesión” permite al usuario cerrar su sesión actual volviendo a la vista de Inicio de Sesión (figura C.1). Por último, el tercer botón etiquetado como “Eliminar Cuenta” permite al usuario eliminar su cuenta, lo cual haría que fuera eliminado del sistema y perdería el acceso a la aplicación. Debido a que la eliminación de la cuenta es un proceso permanente e irrecuperable, antes de proceder a la eliminación se solicita al usuario confirmación (figura C.10).



(a) Vista Perfil (Parte superior) (b) Vista Perfil (Parte inferior)

Figura C.9: En la figura C.9(a) podemos ver parte de la vista de Perfil del Usuario donde puede consultar y cambiar algunos datos de su perfil. En la figura C.9(b) podemos ver parte inferior de la vista de Perfil del usuario donde están los botones de cambio de contraseña, cierre de sesión y eliminación de la cuenta.

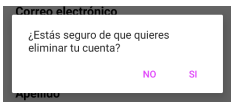


Figura C.10: Diálogo de confirmación de eliminación de la cuenta.

Si el usuario pulsa sobre cualquier evento de la lista en la vista de Eventos del Usuario (figura C.5), la aplicación muestra la vista principal del Evento (figura C.11). Como podemos ver en la parte superior de la vista encontramos una barra de acción, la cúal permite volver a la vista principal del usuario (figura C.5) pulsando la flecha, o desplegar el menú de opciones (figura C.12).

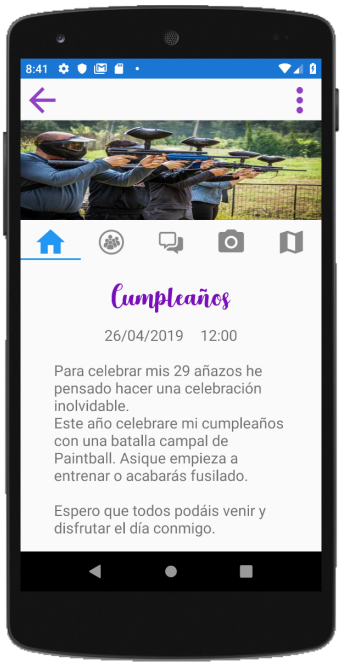


Figura C.11: Vista principal del evento. En esta captura en la vista principal del evento se está mostrando la vista de información del evento.

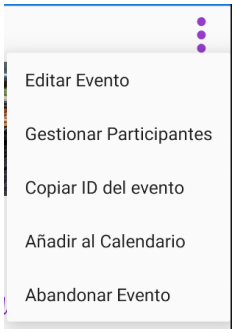
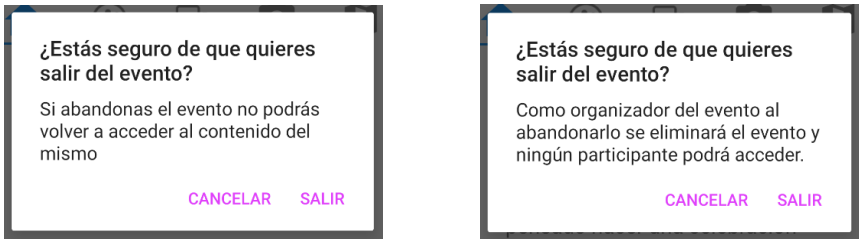


Figura C.12: menú desplegable de Opciones. Da acceso a diferentes funcionalidades y configuraciones del evento.

En el Menú de Opciones (figura C.12), encontramos un máximo de 5 opciones diferentes, las dos primeras solo son visibles para el organizador del grupo mientras que las 3 últimas son accesibles para todos los participantes.

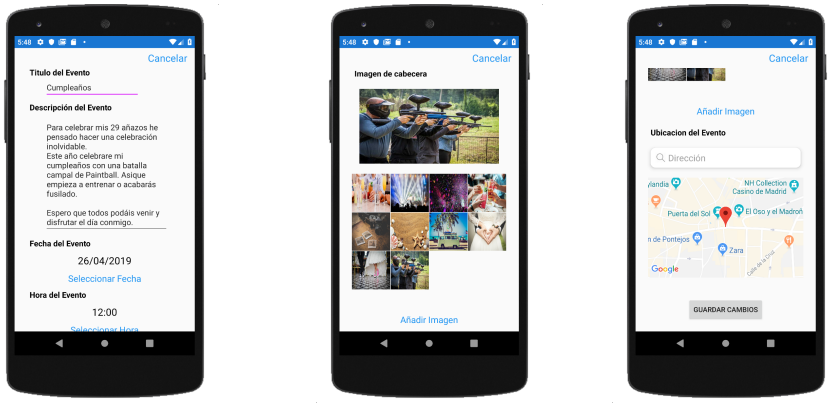
- **Editar Evento (Organizador):** Opción que da acceso a la ventana de edición del evento (figura C.14).
- **Gestionar Participantes (Organizador):** Opción que da acceso a la ventana de gestión de participantes (figura C.15).
- **Copiar ID del Evento (Todos):** Opción que copia en el portapapeles del dispositivo el ID del evento.

- **Añadir al Calendarios (Todos):** Opción que permite añadir el evento a Google Calendar.
- **Abandonar Evento (Todos):** Opción que permite a los participantes abandonar un evento. En el caso de los participantes, el evento es eliminado de los eventos del participante y no podrá acceder de nuevo al evento. En el caso del organizador, el abandono del grupo supondrá la eliminación del evento y ningún participante podrá volver a acceder al evento. Antes de proceder al abandono, el sistema solicita la confirmación del usuario informándole sobre esto (figura C.13).



(a) Confirmación Participante (b) Confirmación Organizador

Figura C.13: En la figura C.13(a) aparece el Diálogo de confirmación de abandono del evento de un participante. En la figura C.13(b) aparece el Diálogo de confirmación de abandono del evento del organizador.

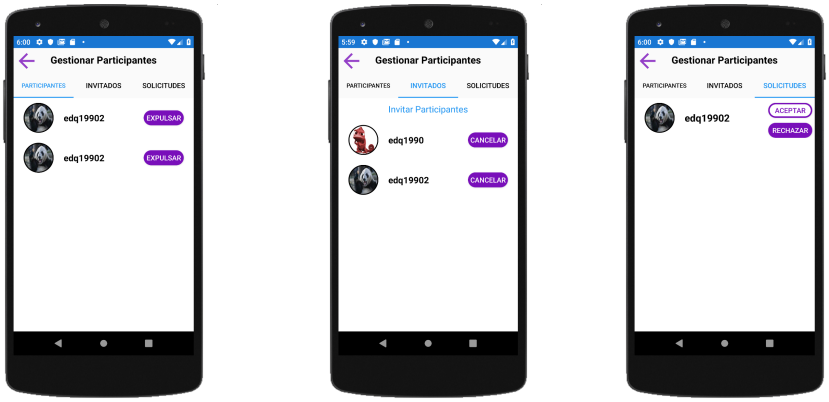


(a) Parte superior (b) Parte media (c) Parte inferior

Figura C.14: En la figura C.14(a) podemos ver la parte superior de la vista de Edición del Evento donde el organizador puede modificar la información del evento (título, descripción, fecha y hora). En la figura C.14(b) podemos ver parte media de la vista de Edición del Evento donde el organizador puede modificar la imagen de cabecera. En la figura C.14(c) podemos ver parte inferior de la vista de la vista de Edición del Evento donde el organizador puede modificar la localización del evento.

La vista de Edición del Evento (figura C.14) permite al organizador modificar toda la información sobre el evento: título, descripción, fecha, hora, imagen de cabecera y ubicación.

La vista de Gestión de Participantes (figura C.15), permite al organizador gestionar los participantes del evento a través de tres pestañas diferentes. La primera le permite ver que participantes hay actualmente en el evento y expulsarlos (figura C.15(a)). La segunda le permite ver las invitaciones pendientes de ser aceptadas, cancelar las invitaciones pendientes y enviar nuevas invitaciones pulsan-



(a) Vista Participantes (b) Vista Invitados (c) Vista Solicitudes

Figura C.15: En la figura C.15(a) podemos ver la primera pestaña de la vista de Gestión de Participante, que se corresponde con la lista de participantes del evento. En la figura C.15(b) podemos ver la segunda pestaña de la vista de Gestión de Participante, que se corresponde con la lista de invitaciones del evento. En la figura C.15(c) podemos ver la tercera pestaña de la vista de Gestión de Participante, que se corresponde con la lista de solicitudes del evento.

do “Invitar Participantes” (figura C.15(b)). En caso de pulsar “Invitar Participantes” se abre una nueva vista que permite al organizador buscar usuarios y enviarles invitaciones (figura C.16). Por último, la tercera pestaña permite al organizador aceptar o rechazar solicitudes de acceso de usuarios al evento (figura C.15(c)).

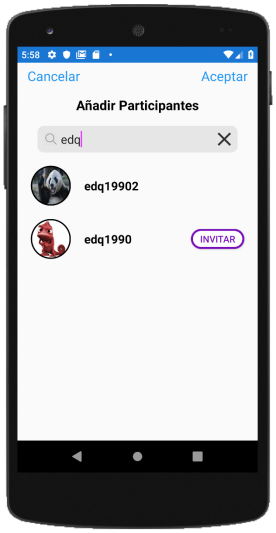


Figura C.16: Vista de Adición de Invitaciones a un evento.

La segunda pestaña de la Vista Principal del Evento es la vista de participantes (figura C.17). En esta vista todos los participantes del evento pueden ver quién más participa en el evento.

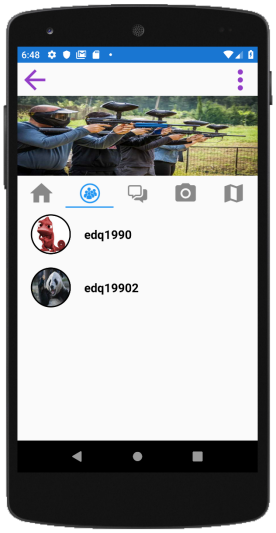
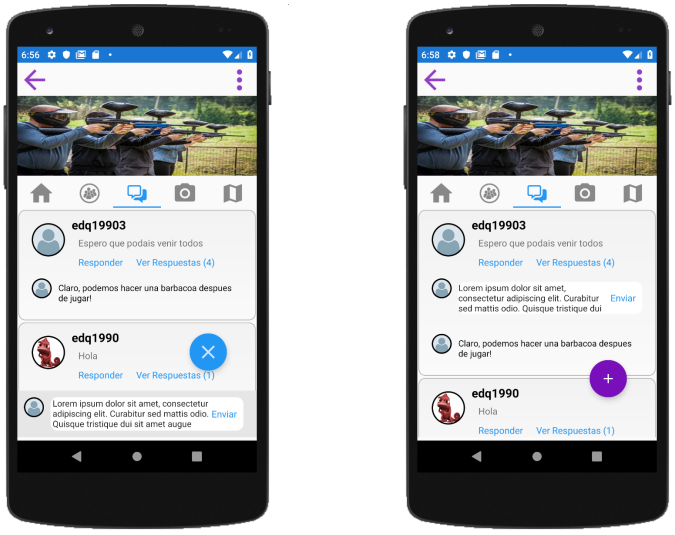


Figura C.17: Vista de Participantes del Evento

En la tercera pestaña de la Vista Principal del Evento encontramos la vista de Mensajes (figura C.18). En esta vista se pueden ver los mensajes escritos hasta el momento en el evento, y añadir nuevos mensajes pulsando sobre el botón con el símbolo "+" y aparecerá una barra de escritura inferior que nos permitirá escribir un nuevo mensaje (figura C.19(a)). Además, si se pulsa en cualquier mensaje sobre el botón "Responder" aparece una barra de escritura en el propio mensaje que nos permite escribir de manera rápida una respuesta a ese mensaje (figura C.19(b)).

Figura C.18: Vista de Mensajes del Evento

En el caso de pulsar sobre el botón "Ver Respuestas" de un mensaje se abrirá la vista de Detallada de Mensaje (figura C.20), en la cual podemos ver el mensaje en la parte superior de la vista y debajo



(a) Añadir mensaje (b) Añadir respuesta

Figura C.19: En la figura C.19(a) se puede observar la barra de escritura inferior que permite añadir un nuevo mensaje al evento. En la figura C.19(b) se puede observar la barra de escritura que aparece debajo de un mensaje al pulsar “Responder” y que permite añadir una respuesta al mensaje.

del mismo una lista con todas las respuestas del mensaje. Además, se dispone de una barra inferior de escritura que permite añadir una respuesta nueva al mensaje.

Figura C.20: Vista de Detallada de un mensaje del Evento

La cuarta pestaña de la vista principal del evento se corresponde con la vista de la Galería del Evento (figura C.21). En esta vista los usuarios pueden ver la galería completa con todas las fotografías subidas por todos los usuarios. Además, si el participante pulsa sobre el botón “+”, se accede a la galería de su dispositivo para seleccionar nuevas imágenes para subir a la galería del evento.

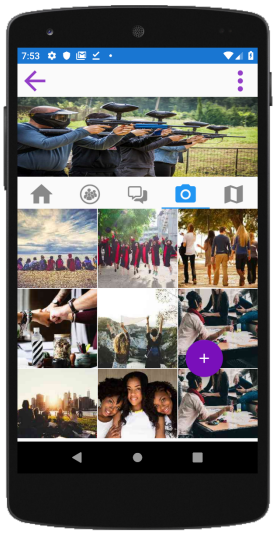
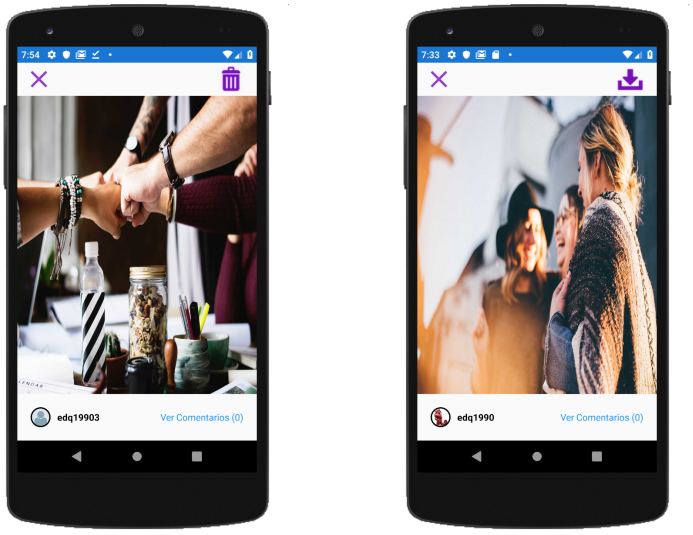


Figura C.21: Vista de la Galería de Fotografías del Evento

Si se pulsa sobre cualquier fotografía del evento, se abrirá la vista detallada de la fotografía (figura C.22), en la cual podemos ver una foto de manera ampliada. En la barra de acción superior encontramos dos botones, el primero (a la izquierda) con el símbolo “X” nos devuelve a la vista de la galería del evento. El segundo botón (a la derecha) será diferente en función de si el usuario que está viendo la foto ha sido el que ha subido la foto a la galería (figura C.22(a)) o por el contrario está viendo una foto subida por otro participante (figura C.22(b)). En el primer caso se muestra un icono de una papelera que permitirá al usuario eliminar una foto subida por el mismo. En caso de estar viendo una foto subida por otro usuario se muestra un icono de descarga que permitirá al usuario descargar la fotografía a la galería de imágenes de su dispositivo.

Después aparece la imagen ampliada que se está visualizando, se puede recorrer la galería haciendo swipe hacia derecha o izquierda, lo cual pasará a la fotografía anterior o posterior, respectivamente.

En la parte inferior de la vista vemos otra barra de información, en este caso se nos muestra el usuario que ha subido la imagen a la galería (a la izquierda) y un botón etiquetado como “Ver Comentarios” que nos lleva a la vista de comentarios de la foto (a la derecha). La vista de comentarios (figura C.23), permite a los participantes ver los comentarios y añadir nuevos comentarios a una fotografía. Para añadir un nuevo comentario solo es necesario introducir el texto en la barra de escritura inferior y pulsar sobre “Enviar”.



(a) Foto detalle propia (b) Foto detalle de otro

Figura C.22: En la figura C.22(a) se puede observar una fotografía en la vista detallada que ha sido subida por el mismo usuario que la está viendo. En la figura C.22(b) se puede observar una fotografía en la vista detallada que ha sido subida por un usuario diferente al que la está viendo.

Figura C.23: Vista de Comentarios de una Fotografía

Cuando en la vista de Galería se pulsa sobre el botón “+”, se lanza una vista del sistema que permite al usuario seleccionar imágenes de la galería de su dispositivo para subir a la galería del evento. Cuando el usuario selecciona una o varias fotografías al pulsar “Abrir” o “Aceptar” (dependiendo del dispositivo), vuelve a la aplicación y se le muestra la vista de confirmación de subida (figura C.24). En esta vista el usuario puede ver las imágenes que ha seleccionado para subir antes de realizar la subida. Además, puede deseleccionar las que no quiera que sean subidas finalmente a la galería del evento. En caso de que el usuario haya seleccionado más de 20 imágenes y pulse Aceptar el sistema le lanza un aviso y le solicita que deseleccione las restantes para que solo haya 20. Si hay menos de 20 imágenes seleccionadas y se pulsa “Aceptar”, comienza el proceso de subida de las fotografías.

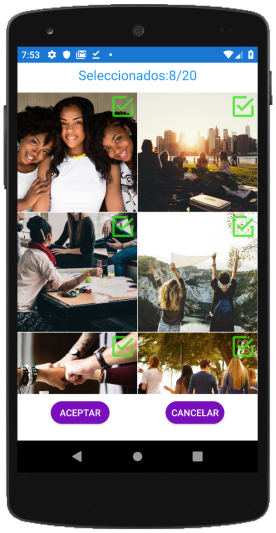


Figura C.24: Vista de confirmación de subida de fotografías a la galería.

Por último, en la quinta y última pestaña de la vista principal del evento encontramos la vista de Ubicación del Evento (figura C.25). En esta vista los participantes pueden ver en un mapa la ubicación en la que se celebrará el evento. Además, en la parte inferior derecha, tiene dos botones que le permiten abrir la navegación de su dispositivo para ir a la ubicación del mapa (izquierda) o abrir la ubicación en la aplicación Google Maps.

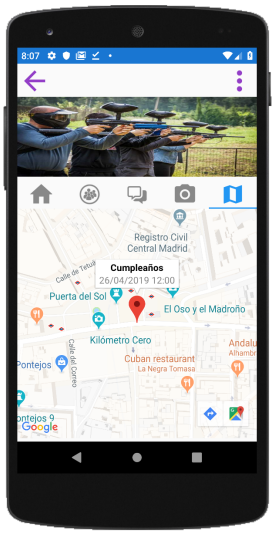


Figura C.25: Vista en la que se muestra la ubicación del evento en un mapa